Des graphèmes à la langue et à la connaissance

Yannis Haralambous

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Université de Bretagne Occidentale

HABILITATION À DIRIGER DES RECHERCHES

L'UNIVERSITE DE BRETAGNE OCCIDENTALE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Informatique

Par
Yannis HARALAMBOUS

Des graphèmes à la langue et à la connaissance

Thèse présentée et soutenue à Brest, le 2 novembre 2020
Unité de recherche : Lab-STICC, UMR CNRS 6285 (équipe DECIDE)

Rapporteurs avant soutenance :

Georges Antoniadis, professeur, Université de Grenoble, France
Christa Dürscheid, professeur, Université de Zurich, Suisse
Éric Saux, Maître de conférences HDR, École navale, Brest, France

Composition du Jury :

Président :	Reinhardt Euler	Professeur émérite, Université de Bretagne Occidentale, Brest
Examinateurs :	Joseph Dichy	Professeur, Université canadienne de Dubaï, Dubaï
	Frédéric Landragin	Directeur de recherche, ENS, Laboratoire Lattice, CNRS, France
	Vincent Rodin	Professeur, Université de Bretagne Occidentale, France

Invité(s)
Annie Gravey Directrice d'études, IMT Atlantique, Brest

HABILITATION À DIRIGER DES RECHERCHES

présentée devant

l'Université de Bretagne Occidentale

par

Yannis Haralambous

Équipe d'accueil :
IMT Atlantique, Laboratoire LabSTICC (DECIDE)

Titre du mémoire :

DES GRAPHÈMES À LA LANGUE ET À LA CONNAISSANCE

Soutenue le lundi 2 novembre 2020 devant la commission d'examen

Georges	Antoniadis	Université de Grenoble, France	Rapporteurs
Christa	Dürscheid	Université de Zurich, Suisse	
Éric	Saux	École navale, France	
Joseph	Dichy	Université canadienne de Dubaï, Dubaï	Examinateurs
Reinhardt	Euler	Université de Bretagne Occidentale, France	
Annie	Gravey	IMT Atlantique, France	
Frédéric	Landragin	ENS, Laboratoire Lattice, CNRS, France	
Vincent	Rodin	Université de Bretagne Occidentale, France	

Table des matières

Introduction

Selon les textes officiels (*Arrêté du 23/11/1988 relatif à l'habilitation à diriger des recherches, version consolidée du 7/8/2020, article 4*),

> Le dossier de candidature comprend soit un ou plusieurs ouvrages publiés ou dactylographiés, soit un dossier de travaux, accompagnés d'une synthèse de l'activité scientifique du candidat permettant de faire apparaître son expérience dans l'animation d'une recherche.

C'est cette synthèse de travaux que ce document se veut être. Il est composé de quatre chapitres. Pour en expliquer le choix et la motivation, voici quelques informations sur mon itinéraire de chercheur.

Après ma thèse ès sciences mathématiques soutenue en juillet 1990 à l'université de Lille 1, je me suis tourné vers la typographie numérique et l'internationalisation du document électronique. Les années 90 furent celles de l'émergence d'Unicode et du Web. Pour ma part, chargé d'enseignement à l'INALCO et en activité libérale, je me suis intéressé à la structure des écritures orientales ainsi qu'aux spécificités des grands textes des « religions du livre » (bible hébraïque, coran, etc.). En ce faisant, j'ai acquis des compétences sur les codages de caractères et les formats de fontes, et j'y ai consacré un pavé de plus de 1 000 pages publié chez O'Reilly France en 2003, et ensuite traduit en anglais et publié chez O'Reilly US en 2007.

En intégrant l'IMT Atlantique (ENST Bretagne à l'époque) en 2001, mes activités de recherche se sont diversifiées :

1. je me suis intéressé à la linguistique et j'ai découvert les travaux de l'école française de graphématique (Jacques Anis, Nina Catach, et autres). Depuis quelques années, une nouvelle branche de la linguistique, dédiée à l'écriture, est en train d'émerger : la *grapholinguistique*. Je me suis investi dans cette discipline : depuis quelques années j'organise, à un rythme biennal, le principal colloque international du domaine, et je suis éditeur-en-chef d'une série d'ouvrages spécialisée dans ce domaine.
   C'est pour ces raisons que le premier chapitre de ce mémoire est dédié à la grapholinguistique. Plutôt que d'en donner une introduction générale, j'ai préféré traiter un sujet spécifique lié à l'informatique et qui représente bien mes travaux des trente dernières années : une discussion critique du codage de caractères Unicode sous le prisme théorique de la grapholinguistique. Cette discussion est suivie d'une description des *textèmes* (une notion généralisant les graphèmes et les graphes, introduite dans les années 2005) et d'un résumé de mes travaux en sinographématique.
2. Mon passé de mathématicien m'a incité, dès mon intégration à l'IMT Atlantique, à m'intéresser à la logique et à ses applications en intelligence artificielle. En particulier, je me suis intéressé aux méthodes de représentation des connaissances que j'ai utilisé dans deux domaines : (a) la modélisation des figures de la géométrie euclidienne avec un objectif de fouille et (b) l'inter-

action homme-machine avec une base de connaissances du Service hydrographique et océanographique de la Marine. Une synthèse de ces deux applications constitue le deuxième chapitre de ce mémoire.

3. Enfin, en intégrant l'équipe de recherche DECIDE du laboratoire LabSTICC je me suis intéressé au *machine learning* et me suis spécialisé, tout naturellement, à la fouille de texte, une discipline au croisement de la linguistique, de l'informatique et des mathématiques. Ce domaine n'a cessé d'évoluer ces vingt dernières années et est toujours riche en défis et en applications. Dans le troisième chapitre de ce mémoire je décris trois projets que j'ai pu mener : (a) une série de travaux sur la mesure de la similarité sémantique en utilisant Wikipédia en tant que corpus d'apprentissage, (b) un projet sur la modélisation de la recherche en ligne et la navigation sérendipitaire sur le Web et, enfin, (c) un travail en cours : l'interaction entre interjections/pauses et arbre syntaxique dans le but de détecter des comorbidités psychiatriques.

À ces trois chapitres, portant sur des axes de recherche complémentaires, s'ajoute un quatrième chapitre intitulé « Exercices de vulgarisation ». Il s'agit de deux tentatives de vulgarisation de sujets particulièrement difficiles d'accès à l'intention d'un public assez vaste (la revue *Quadrature* cible les étudiants des deux premiers cycles mais aussi les lycéens mordus de mathématiques et les professeurs de mathématiques du secondaire ou des classes préparatoires), dans l'espoir de susciter des vocations chez les lecteurs. Ma méthode, que je décris dans ce chapitre, consiste à alimenter leur motivation par le défi. Les sujets choisis sont (1) la sémantique formelle de Montague et (2) la logique combinatoire à travers la forêt enchantée de Smullyan. Après un bref état de l'art sur les aspects didactiques de la vulgarisation des mathématiques en France, je relate dans ce chapitre mes efforts pour maintenir le fragile équilibre qui caractérise la vulgarisation, entre le trop dire et le ne pas dire assez.

Chaque chapitre se termine par une section de perspectives indiquant des projets en cours ou à venir dans le domaine concerné.

Le mémoire se termine par deux listes de références : d'abord celles citées dans le texte et ensuite la liste complète de mes publications et co-publications. Pour éviter la répétition incessante de mon nom, je l'ai remplacé par des initiales.

Chapitre 1

Le texte informatique

L'homme écrit. Tel fut l'*incipit* de [YH, 2004a], ouvrage basé sur la symétrie entre deux visions de l'écriture informatique : les *caractères*, une abstraction destinée à l'échange d'information, et les *glyphes*, instances d'images servant à la lecture par l'humain. Or, l'écriture est une modalité du *langage naturel*, et celui-ci est l'objet d'étude de la linguistique. Nous appellerons *grapholinguistique* la branche de la linguistique qui s'intéresse à la modalité écrite. Cette branche est, hélas, controversée par les pères de la linguistique moderne, Ferdinand de Saussure et Leonard Bloomfield [Dürscheid, 2016, p. 18]. Dans ce chapitre il sera question de texte informatique du point de vue de la grapholinguistique. Nous parlerons d'abord du codage Unicode qui, après quelques décennies chaotiques où chaque pays, chaque système d'exploitation et chaque fournisseur de matériel informatique codait le texte à sa manière [YH, 2004a, p. 27–50], s'est imposé aujourd'hui comme le seul codage de texte universel, au niveau mondial.

1.1 Unicode d'un point de vue linguistique

Unicode est un standard industriel de représentation et de transmission de données textuelles. Il a été créé en 1991 et est aujourd'hui à sa 13^{e} version. Unicode s'appuie sur des principes de base mais est également obligé de se soumettre à la réalité technologique et industrielle et à des considérations politiques.

Il est le premier codage textuel dans l'histoire de l'informatique qui *instruit* l'utilisateur désireux de communiquer à travers différents systèmes d'écriture. Le Consortium Unicode publie lors de chaque mise-à-jour du codage un ouvrage de plus de mille pages [The Unicode Consortium, 2020] décrivant l'utilisation optimale d'Unicode pour les différents systèmes d'écriture, ainsi que quelques dizaines d'annexes et rapports traitant de problèmes tels que la césure, le rendu bidirectionnel, le texte vertical, les émojis, etc.

Pendant la trentaine d'années de son existence, le Consortium Unicode a patiemment mis en place un vocabulaire technique pour décrire les concepts informatiques et grapholinguistiques afférents. Pour décrire l'approche Unicode de l'écriture, nous allons comparer deux processus : celui d'une personne lisant (et comprenant) un texte affiché sur un dispositif électronique et celui d'une machine « lisant » et analysant un texte provenant d'un flux d'entrée. Dans les deux cas, le but est la « compréhension » du texte, dans le sens de l'accès aux différents niveaux linguistiques, y compris le niveau sémantique.

La raison pour laquelle nous nous proposons de comparer ces deux processus, qui pourtant

semblent fondamentalement différents, est le fait qu'ils dévoilent la double nature d'Unicode : pour que l'humain soit capable de lire du texte sur un écran, Unicode (assisté par le moteur de rendu du système d'exploitation et par les polices de caractère auxquelles le système a accès) doit fournir les informations nécessaires pour que le rendu du texte soit possible ; d'autre part, pour que les algorithmes « lisent » le texte à partir du flux d'entrée, Unicode doit fournir les informations nécessaires pour que toutes les strates d'information linguistiques soient acheminées. Ces deux besoins d'information sont complémentaires et Unicode a été développé de façon à les satisfaire simultanément.

1.1.1 Préliminaires linguistiques

Quand Martinet [1970] définissait la *double articulation*, il considérait les phonèmes en tant que niveau le plus bas d'articulation dans la hiérarchie du langage. Un phonème est une unité distinctive dans un système et n'a pas de sens en soi. Le sens émerge lorsqu'on aligne les phonèmes en suites, suites que l'on appelle des *morphèmes*. Ce processus est appelé *deuxième articulation*. Pour vérifier que les phonèmes sont bien des unités distinctives, Martinet utilise la méthode de *commutation* : si en remplaçant une unité par une autre le sens du morphème change, alors ces deux unités méritent bien le statut de phonème ; si non, ce ne sont guère que des allophones du même phonème (en guise d'exemple, comparer, en français, le « r roulé » (consonne roulée alvéolaire voisée) et le « r grasseyé » (consonne roulée uvulaire voisée), les deux connotent des origines géographiques ou sociales, mais ne dénotent pas de changement de sens). D'autre part la formation de suites de morphèmes donne accès, à travers la syntaxe, à des niveaux supérieurs de sens — ce processus est appelé *première articulation*.

Anis [1988a, p. 89] et Günther [1988, p. 77] utilisent la même méthode pour définir les *graphèmes* : dans leur approche, appelée *approche autonomiste* (cf. Glück [1987, p. 68–74]) car indépendante de la phonologie, les graphèmes sont des objets de la langue écrite qui n'ont pas de sens en soi mais constituent des unités distinctives du système. Ils sont définis par commutation[1] et en les concaténant (opération définie dans [Sproat, 2000, p. 13]) le sens émerge : les graphèmes <b>, <u>, <t> et <s> n'ont pas de sens quand ils sont considérés individuellement, mais leur concaténation <buts> a un sens et on y décèle deux morphèmes : le morphème lexical « but » et le morphème grammatical « s », dénotant le pluriel.

La propriété d'unité distinctive est nécessaire parce que nous vivons dans un monde analogique. Quand une infinité continue de sons peut être perçue par l'oreille humaine et une infinité continue de formes peut être perçue par l'œil humain, le cerveau doit d'abord sélectionner ceux qui relèvent de la communication linguistique (appelés *phones*, resp. *graphes*) et ensuite les classer dans un nombre fini de classes, par la concaténation desquelles les morphèmes sont créés. Les *graphes* sont étudiés par la discipline émergente de *graphétique* [Meletis, 2015], dont le nom est inspiré par la discipline analogue de *phonétique*, dans le cas de la modalité orale de langue.

Mis à part les *graphes* (formes utilisées dans la communication écrite) et les *graphèmes* (unités linguistiques élémentaires), Rezec [2009] introduit la notion intermédiaire de *forme de base*, autrement dit de cluster dans l'espace des graphes représentant le même graphème, tel que $\left|\text{ɑ}\right|$ et $\left|\text{a}\right|$, qui sont des allographes du graphème <a>. À noter que les notions de forme de base ou de graphème sont liés à une langue ou à un système de notation donnés : ainsi |ɑ| et |a| représentent le même graphème en français mais des graphèmes différents dans l'alphabet phonétique international (le <ɑ>

[1] Cette définition concorde avec la définition institutionnelle des vingt-six lettres de l'alphabet latin utilisées en français, mais génère des incertitudes pour les lettres diacritées ou spéciales. Ainsi la ligature <œ> est difficilement remplaçable par un autre signe pour que <œ> mérite le statut de graphème.

représente la voyelle basse postérieure non arrondie, et le <a> la voyelle antérieure correspondante) ou dans certaines langues africaines comme le medumba. De même les graphes $\left|\,\rho\,\right|$ et $\left|\,\varrho\,\right|$ qui sont allographes en grec moderne peuvent représenter des symboles mathématiques différents.

1.1.2 Encodage du texte

Le *Cambridge Dictionary of English* définit le verbe *to encode* comme

> mettre une information dans une forme dans laquelle elle peut être sauvegardée et qui peut être lue uniquement en utilisant des technologies ou des connaissances spéciales.

Cette définition implique trois actions : mettre, sauvegarder et lire. Dans notre contexte, l'acte de « mettre » sera considéré comme « convertir de l'information analogique en une forme numérique », ou « produire de l'information (numérique) sur un support numérique » et nous allons nous restreindre à des données textuelles, où le terme « texte » est pris dans un sens plutôt général, incluant des données dans différents systèmes de notation comme les formules mathématiques, etc. L'action de « sauvegarde » peut être numérique ou analogique (par exemple l'impression papier) et celle de « lecture » peut prendre différentes formes, selon l'acteur : un humain peut lire un texte analogique (optique ou haptique) produit par des moyens mécaniques ou alors un texte analogique produit par un dispositif numérique (utilisant de l'information numérique), une machine peut « lire » un texte analogique (par OCR) ou alors un texte numérique dans le sens d'un programme qui reçoit les données à travers un flux d'entrée.

Nous allons parler de processus de « lecture » dans la section suivante. Pour le moment, considérons la forme dans laquelle les données textuelles sont converties en tant que résultat du processus d'encodage. Le texte en langage naturel (qui est le terrain de prédilection du standard Unicode) est un objet complexe composé de plusieurs strates d'information. Même si nous nous restreignons aux strates de nature linguistique, le processus d'encodage peut prendre différentes formes :

1. Un des dispositifs d'entrée les plus communs est le clavier d'ordinateur, qui est fonctionnellement un descendant de la machine d'écrire. « Encoder » un texte sur une machine à écrire revient à le saisir. Saisir un texte revient à choisir des touches, à les appuyer et à obtenir une séquence graphétique 1-dimensionnelle [Meletis, 2019, p. 117–120] sur support papier. La largeur du papier étant limitée, le retour-chariot de la machine à écrire permet au scripteur de produire une séquence graphétique 2-dimensionnelle dans l'espace de la page. Le clavier d'ordinateur possède également une touche « retour-chariot » mais son utilisation n'est pas obligatoire puisque la mémoire de l'ordinateur peut être considérée comme une « page de largeur infinie » et donc l'« encodage » d'un texte à travers le clavier d'ordinateur produit une longue séquence 1-dimensionnelle d'unités élémentaires d'information correspondant aux touches (ou combinaisons de touches) appuyées par le scripteur. Le résultat de ce type d'« encodage » est un objet numérique appelé *texte brut* et c'est le type de données qu'Unicode se propose d'encoder.
2. D'autres techniques traditionnelles de production de texte, telles que la typographie, ont un spectre plus large de méthodes de communication (italiques, interlettrage, couleur, etc.) qui peuvent endosser différentes fonctions linguistiques et paralinguistiques. Elles peuvent être considérées comme partie intégrante du texte et doivent également être encodées. Les langages de balisage comme XHML ou XSL-FO gèrent ce type d'encodage efficacement, le résultat est appelé *texte riche*.

3. Le langage naturel a deux modalités principales[2] : les modalités écrite et orale. Dans les langues à orthographe superficielle comme l'italien ou l'arabe standard pleinement voyellé, on peut facilement convertir les données entre les deux modalités, avec peu ou pas de perte d'information ; par contre, dans les langues à orthographe profonde, telles que l'anglais ou le grec, ce processus requiert des algorithmes élaborés et des lourdes ressources linguistiques. En annotant phonétiquement un texte (encodé), on peut avoir un accès immédiat et simultané aux deux modalités. Ainsi, un texte encodé dans le format FoLiA [van Gompel et Reynaert, 2013] peut être muni d'annotations phonétiques et/ou phonologiques.
4. Étant un objet linguistique, le texte peut être analysé en utilisant des méthodes linguistiques traditionnelles et les résultats de cette analyse peuvent être inclus dans le texte, on obtient ainsi du *texte annoté*. Cela peut sembler inutile pour un lecteur humain qui connaissant la langue du texte, mais peut être utile pour un lecteur humain qui apprend la langue ou pour une machine qui devrait autrement procéder aux mêmes analyses. Ainsi, la première étape de la plupart des algorithmes de traitement automatique de la langue est une analyse morphosyntaxique. L'obtention d'une représentation du texte possédant cette information, par exemple dans le format CoNLL-U [Marneffe et al., 2013], est un autre type d'« encodage » de texte, incluant les parties du discours, les lemmes et les relations de dépendance, de manière explicite.
5. Mais pourquoi s'arrêter au niveau syntaxique ? L'étape qui suit est celle de l'annotation sémantique, où l'on encode les concepts et les relations entre eux, en les alignant avec des ontologies, des bases de connaissances ou d'autres ressources sémantiques. Cela est possible, par exemple, à l'aide de technologies du Web sémantique telles que OWL et RDF se servant du langage de balisage XML. Le texte encodé de cette manière est optimalement traité par les algorithmes de traitement automatique de la langue.

Nous voyons donc que l'« encodage » de texte peut être plus ou moins élaboré et riche en information selon le « lecteur » ciblé. Quand le « lecteur » est un humain, alors les approches 1 et 2 sont clairement distinctes des approches 3–5. En effet, les approches 1 et 2 produisent un résultat visuel qui peut être lu par un humain, alors que les approches 3–5 enrichissent le texte en ajoutant des informations supplémentaires. Quand le « lecteur » est la machine, il n'y a pas d'étape visuelle et la distinction entre 1, 2 et 3–5 est sans objet.

Plusieurs corpus importants se servent de plus d'une méthode d'encodage. Ainsi le *Digital Corpus of Sanskrit* [Hellwig, 2019] est un objet numérique qui peut être « lu » par un humain de manière traditionnelle, mais qui contient également des informations morphologiques et lexicales. Ces informations peuvent être communiquées à l'utilisateur humain à travers une interface homme-machine dédiée, ou alors être « lues » directement par les algorithmes de traitement automatique de la langue qui traitent le texte. Le corpus *Quranic Arabic Corpus* [Dukes et al., 2013] comporte également des dépendances syntaxiques et des annotations sémantiques pour la totalité du texte coranique.

Parfois les frontières entre les technologies mentionnées deviennent floues. En guise d'illustration, voici deux exemples impliquant Unicode :

- En japonais et en chinois, la méthode *ruby* consiste à ajouter la réalisation phonétique des morphèmes (écrits en caractères kanji/hanzi) en utilisant des syllabogrammes kana de petit corps, placés au-dessus des sinogrammes, comme dans 会社(かいしゃ) (« société », prononcé かいしゃ *kaisha*). Alors qu'Unicode proclame qu'il ne code que du texte brut, il propose néanmoins trois

[2] Sans parler de la gestualité du langage des signes.

opérateurs d'annotation interlinéaire pour indiquer le début d'une séquence de base, le passage entre base et annotation et la fin de la séquence d'annotation. XHTML propose également des balises pour l'annotation interlinéaire (l'élément `ruby`) et cette méthode est recommandée par le W3C, plutôt que les caractères Unicode correspondants (cf. [Sawicki et al., 2001] et [Dürst et Freytag, 2000]). En tant qu'annotation, le ruby est essentiellement phonologique et morphologique[3] puisque les bases d'annotation sont des morphèmes : de ce fait, le ruby se place entre les approches 3 et 4.

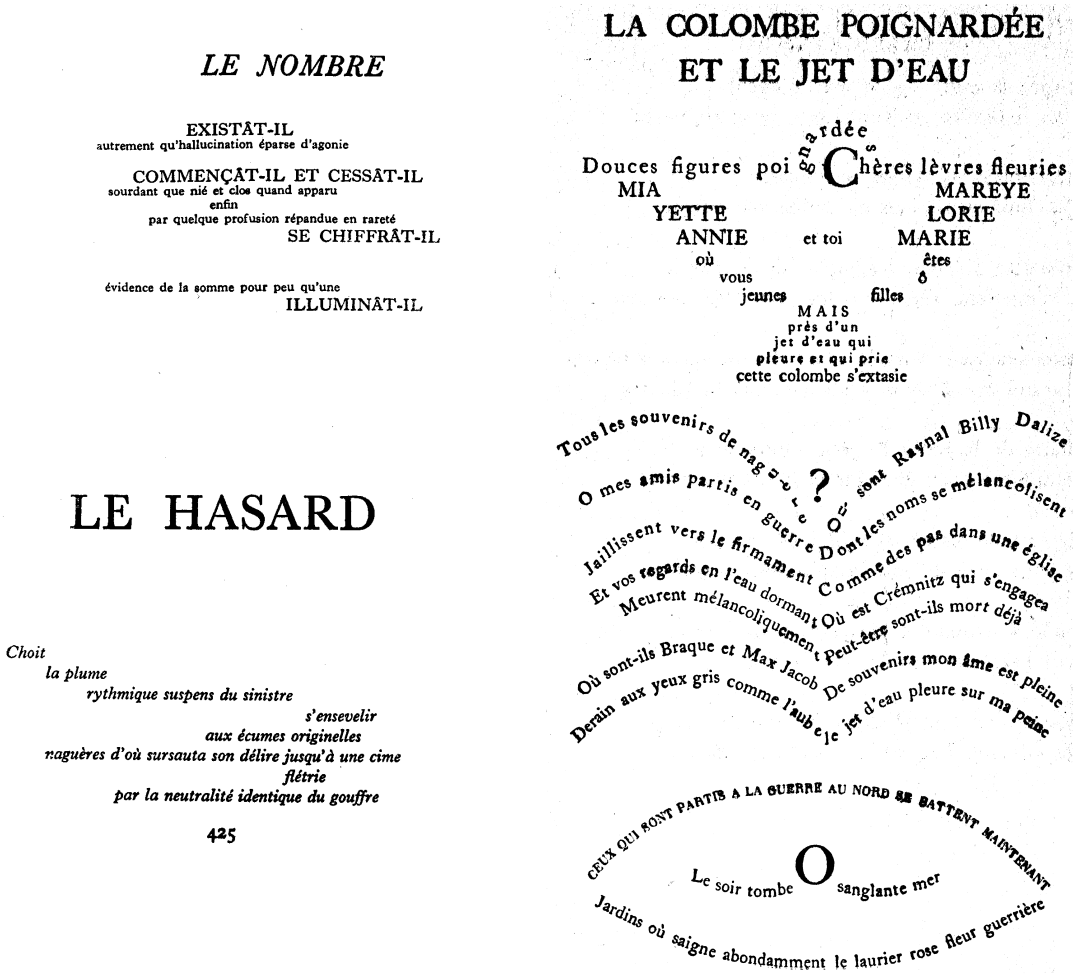
- Il existe des caractères Unicode sans représentation visuelle [YH, 2004a, p. 98–102]. Ceux-ci sont porteurs d'information de nature morphologique, syntaxique ou sémantique : le *soft hyphen*, caractère de césure potentielle, indique les frontières des syllabes en vue de la segmentation des séquences graphétiques 1-dimensionnelles ; l'*invisible separator* marque des caractères consécutifs comme faisant partie d'une liste, ce qui en fait une information d'ordre syntaxique ; l'*invisible times* marque des caractères consécutifs comme étant des symboles mathématiques participant à une opération de multiplication, il a donc un rôle sémantique. L'information portée par les deux derniers peut également être représentée par du balisage dans le langage MathML [Carlisle et al., 2014] : les éléments `apply` et `times`.

L'approche standard pour produire du texte écrit, décrite dans [Meletis, 2019, 117–120], est de concaténer des graphes pour former des séquences 1-dimensionnelles et remplir ainsi l'espace linéaire, jusqu'à atteindre la limite de la partie imprimable de la page et de continuer ensuite sur la ligne suivante. Cette approche, qui est l'approche traditionnelle de l'imprimerie, est utilisée par Unicode et par les moteurs de rendu. Elle assume implicitement que la disposition géométrique des séquences graphétiques 1-dimensionnelles (tant qu'il n'y a pas de structure supérieure 2-dimensionnelle telle qu'une liste ou une table) ne porte pas d'information syntaxique ou sémantique spécifique.

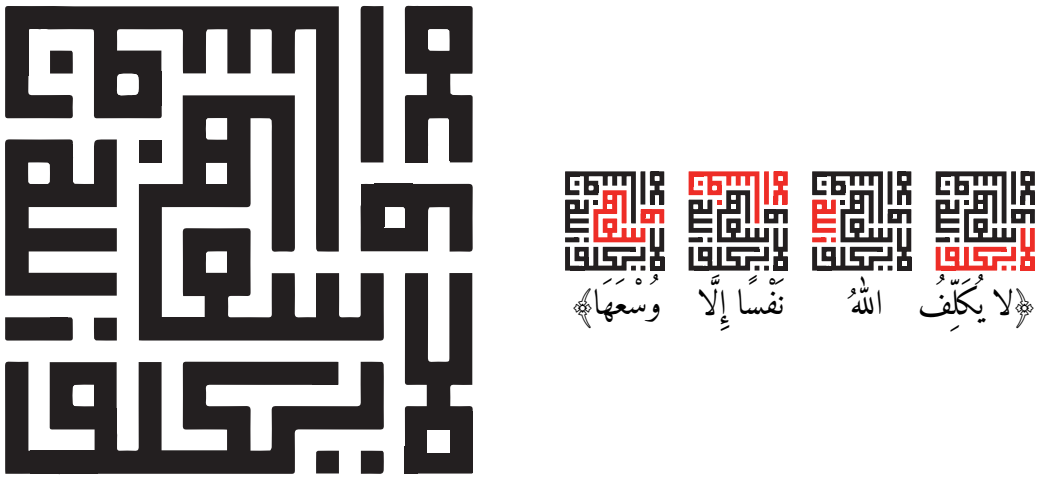
Il existe des cas où la créativité humaine a transcendé ce modèle et nous allons présenter trois exemples (cf. fig. 1.1). Il est légitime de se poser la question si ces cas peuvent être « encodés » par la machine sans perte d'information. Ils sont les suivants :

1. une page de « Un coup de dés jamais n'abolira le hasard » de Mallarmé, où le processus de lecture est spatialement et temporellement structuré par les espaces horizontaux et verticaux, le corps, le choix de fonte et la casse. Pour obtenir le résultat visuel souhaité par Mallarmé, XHTML et XSL-FO ne suffisent pas et un langage de balisage pour décrire des graphiques bidimensionnels et mixte vectoriel/bitmap, tel que SVG [Bellamy-Royds et al., 2018] est nécessaire ;
2. le calligramme « La colombe poignardée et le jet d'eau » d'Apollinaire, dans lequel non seulement les graphes forment un contour graphique, mais le texte et l'image sont également en interaction : par exemple, le caractère doux et immaculé des ailes de la colombe est renforcé par les fragments de texte sur leurs contours : « douces figures » et « lèvres fleuries » et par les six prénoms féminins suivis de la question « où êtes-vous ô jeunes filles ». Également, la plaie de la colombe poignardée est formée par les mots « et toi ». Ici aussi, un langage de balisage tel que SVG est nécessaire pour placer des graphes sur des tracés curvilignes en maintenant la linéarité du texte du poème, et pour encoder les correspondances entre parties du graphique et segments de texte. On pourrait même considérer une description hiérarchique abstraite du

[3] Dans un article à paraître sur les méthodes graphétiques et graphémiques de la fiction spéculative, nous décrivons un autre type de ruby, nommé *atéji*, qui transcende le caractère phonologique traditionnel du ruby en instaurant une relation discursive entre base et annotation.



(a) (b)



(c)

FIG. 1.1: Trois exemples où le corps, le style, la position et la forme des séquences graphétiques participent à la production du sens.

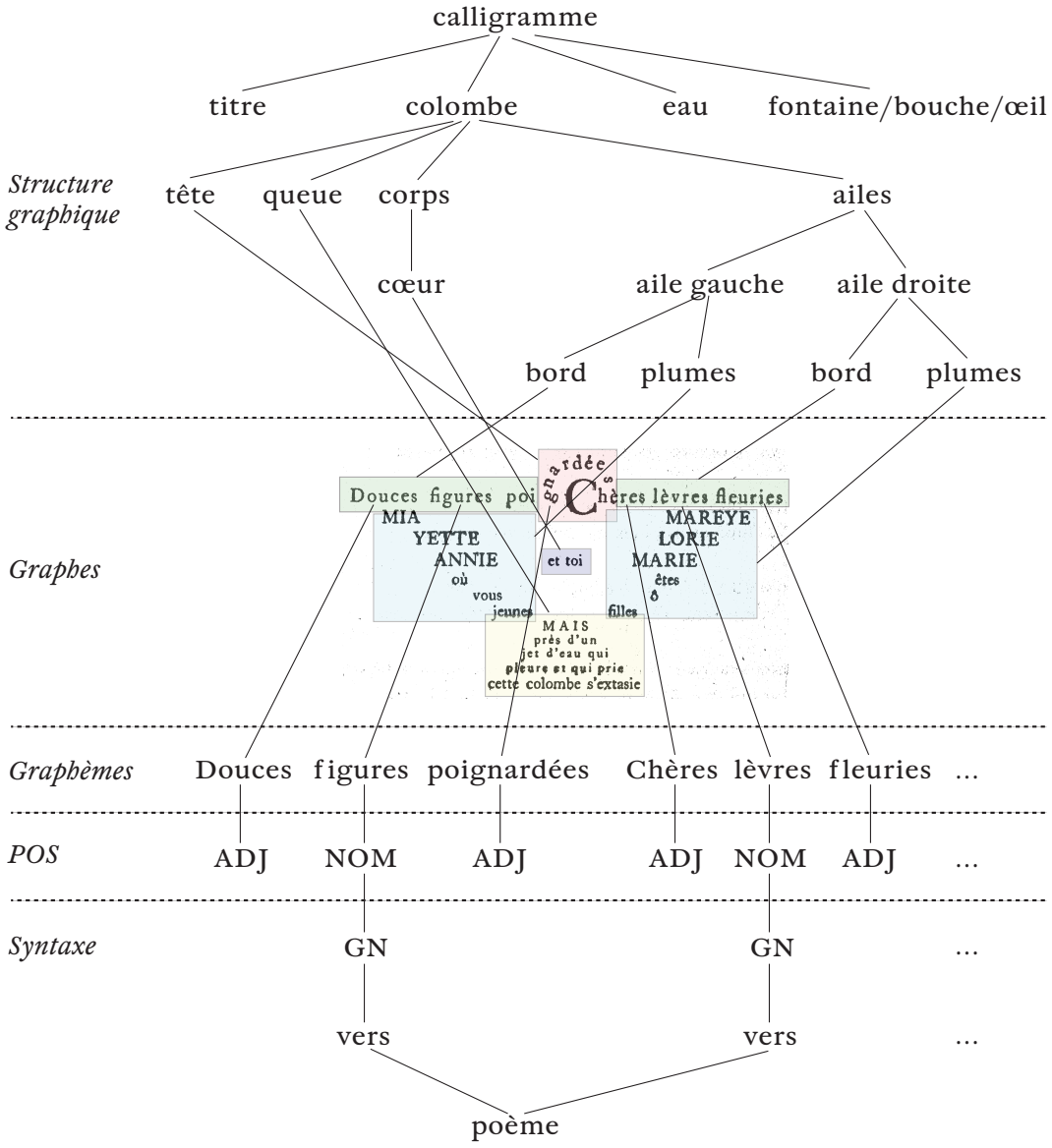


FIG. 1.2: Niveaux d'analyse d'un petit fragment du calligramme « La colombe poignardée et le jet d'eau » d'Apollinaire.

tracé (incluant la colombe, sa tête, ses ailes, sa queue, les plumes, etc.) où chaque élément est lié à un segment de texte, de manière à ce que le poème hérite de la structure graphique du tracé et à ce qu'on ait un alignement entre les structures hiérarchiques graphique et linguistique (cf. fig. 1.2 pour un petit extrait d'une telle série de correspondances) ;

3. et enfin une calligraphie coufique d'un vers coranique : ﴿لا يُكَلِّفُ اللهُ نَفْسًا إِلَّا وُسْعَهَا﴾ (« Allah n'impose à aucune âme une charge supérieure à sa capacité » 2:286), écrit sous forme de spirale démarrant en bas à droite et en avançant dans le sens trigonométrique inverse de manière à ce que le dernier mot se trouve au milieu du tracé. Ici, le fait de reconnaître le texte dans le labyrinthe symbolise, dans le cadre de la religion musulmane, la découverte de la parole de Dieu dans le monde, qui est son livre.
   De nouveau un langage tel que SVG est nécessaire pour que la calligraphie conserve sa une nature duale texte/image, mais aussi une fonte munie de glyphes dynamiques dont les instances sont obtenues à partir de « métagraphes » pour l'énoncé textuel spécifique. [Bayar et Sami, 2010 ; André et Borghi, 1990].

Dans la suite de ce chapitre nous allons adopter une approche orientée tâche et examiner Unicode dans le cadre de trois processus de lecture qui diffèrent par leurs acteurs : l'humain ou la machine.

1.2 Trois processus de « lecture »

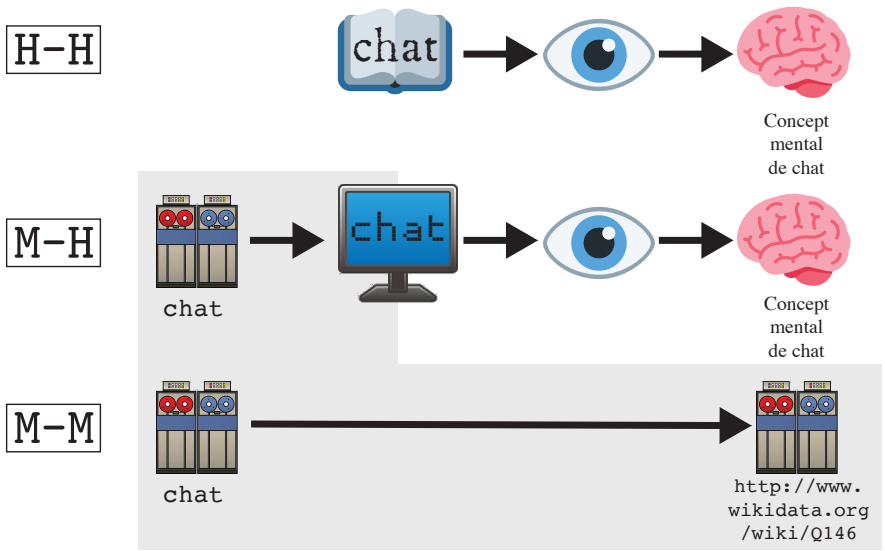


FIG. 1.3: Les processus `H-H`, `M-H` et `M-M` impliquant le morphème <chat> et le concept de chat. Le fond gris dénote le monde numérique.

La discipline de *graphétique perceptuelle* [Meletis, 2015] étudie l'influence de la matérialité de l'écriture sur la perception, la reconnaissance et la lecture. Il y a eu de nombreuses études sur le cas particulier de la graphétique perceptuelle sur les dispositifs électroniques (écran d'ordinateur, tablette, smartphone, etc.). Dans ce domaine, le texte affiché est considéré comme point de départ

et les objets étudiés sont principalement la perception humaine du signal émis par la machine et les processus cognitifs impliqués dans la reconnaissance et la compréhension de données textuelles.

Nous allons étendre le processus de « lecture » à la situation où l'émetteur et le récepteur sont des machines et le canal est purement numérique (de manière à ce qu'on ne nécessite pas de médiation optique). Nous décrivons trois situations où le texte est « lu », correspondant à ces trois processus, illustrés par la fig. 1.3 où le texte est le morphème « chat » :

- `H-H` (« humain → humain ») est le processus de lecture sur support papier (manuscrit ou imprimé) : l'œil perçoit des graphes sous forme d'encre sur la surface du papier, le cerveau reconnaît les graphèmes, les combine en morphèmes et accède aux concepts mentaux qu'ils représentent ;
- `M-H` (« machine → humain ») est le même processus, mais cette fois la surface de lecture est un moniteur d'ordinateur ou autre dispositif électronique. La chaîne `chat` a été enregistrée dans la mémoire de l'ordinateur, codée en Unicode, elle est transmise à un moteur de rendu qui extrait des données d'une fonte, génère une image et la transmet au dispositif d'affichage. À la frontière entre monde numérique (fond gris sur l'image) et monde analogique, le dispositif affiche le mot. Le reste du processus `M-H` est identique à `H-H` ;
- `M-M` (« machine → machine ») est le processus d'accès au même concept à travers des algorithmes de traitement automatique de la langue. L'entrée est la même : le mot `chat` codé en Unicode. Les algorithmes de TAL doivent (a) détecter la langue courante ; (b) utiliser cette information pour détecter des morphèmes/mots ; (c) utiliser le contexte et des ressources linguistiques pour trouver les parties du discours ; (d) désambiguïser : s'agit-il du félin ? ou de l'instrument qui sert à inspecter des canons ? ou du jeu enfantin ? Une fois désambiguïsé le concept est représenté par un IRI (*Internationalized Resource Identifier*, [Dürst et Suignard, 2005]) qui pointe l'une entrée de la base de connaissances Wikidata `http://www.wikidata.org/wiki/Q146`. Cette entrée correspond au concept de chat (animal félin).[4]

Le diagramme de la fig. 1.3 n'est qu'une simplification du processus réel — son but est d'illustrer la dualité des objectifs d'Unicode, qui se doit de fournir suffisamment d'informations au moteur de rendu pour qu'il affiche correctement les graphèmes sur le dispositif et que le processus `M-H` aboutisse ; mais aussi de fournir suffisamment d'informations aux algorithmes de TAL pour qu'ils mènent à bien leur tâche.

Dans le processus `M-M`, les algorithmes de traitement automatique de la langue doivent obtenir en entrée suffisamment d'informations pour accéder à toutes les facettes de l'objet linguistique de départ. Unicode a été crée dans le but de permettre de genre de processus. Mais il doit également permettre le processus `M-H` : dans ce cas, le système transmet la chaîne de caractères Unicode à un moteur de rendu qui va se servir d'une fonte qui va associer les caractères à des images (appelées glyphes) affichées par le dispositif.

Le plus souvent le choix de fonte impliquée dans le processus de rendu dépend du domaine d'application (par exemple, en informatique et en mathématiques les publications sont composées en Computer Modern, la plupart des autres publications scientifiques en Times) ou de la signature graphique de l'éditeur, ce qui fait que la contribution du choix de fonte à la production de sens est secondaire et principalement connotative [Blanchard, 1980, chap. 7]. Pour cette raison, Unicode

[4] Nous avons choisi Wikidata en tant qu'exemple, mais d'autres bases de connaissances existent, telles que WordNet ou Yago.

n'encode pas les fontes et cette information doit être fournie par des protocoles supérieurs, tels que des langages de balisage[5] ou des feuilles de style. De nouveau il existe des cas où la créativité humaine a transcendé cette convention et a élevé le choix de fonte au statut de facteur important dans la production de sens. Le lecteur peut voir dans la fig. 1.4 l'exemple d'une publicité autrichienne : « Gehen Sie wählen ! Andere tun es auch. » (« Allez voter ! D'autres le font aussi »), où le passage en écriture gothique dans la deuxième phrase introduit une connotation d'orientation politique d'extrême-droite[6] attachée en particulier au pronom « d'autres ».

FIG. 1.4: Publicité autrichienne : « Allez voter ! D'autres le font aussi. » (également dans [Dürscheid, 2016, p. 232] et [Schopp, 2008]).

Le choix de fonte ou la disposition spatiale du texte sont opérationnels dans les processus `H-H` et `M-H` mais, à la connaissance de l'auteur, les algorithmes de traitement automatique de la langue ne sont pas encore en mesure d'extraire le sens produit par les propriétés géométriques ou graph(ét)iques — au contraire ils se basent plutôt sur le modèle de « texte brut ».

Ce qui nous ramène aux deux principales questions de ce chapitre :

1. Comment Unicode modélise-t-il l'écriture de manière à gérer et le processus `M-H` et le processus `M-M` ?
2. Quelles sont les notions fondamentales d'Unicode et de quelle manière sont-elles liées aux notions et processus linguistiques ?

Dans les sections qui suivent nous allons explorer les notions Unicode de caractère, catégorie de caractère, glyphe et chaîne de caractères, et les confronter aux notions linguistiques de graphème, classe de graphème, graphe, forme de base, séquence graphé(m|t)ique 1-dimensionnelle, etc.

[5] Le langage SVG [Bellamy-Royds et al., 2018] non seulement permet le choix de fonte, mais propose même des balises XML pour décrire des fontes entières qui peuvent être stockées en mode local ou distant.

[6] Ce qui en réalité est plutôt ironique, puisque c'est Hitler qui a interdit l'écriture gothique en 1943, cf. [YH, 1991e].

1.3 Unicode

1.3.1 Caractères vs. graphèmes

L'unité atomique d'Unicode est le *caractère*. Ce terme a ses racines dans les codages de caractères des années 60 (FIELDATA en 1960, ASCII in 1963, cf. [Mackenzie, 1980] et [YH, 2004a, p. 27–33]). À l'époque, un « caractère » était un « motif de bits spécifique et un sens qui lui est assigné » (*specific bit pattern and an assigned meaning*). Le « sens » était soit un « sens de contrôle » (faire sonner une clochette, supprimer le caractère précédent...) ou un « sens graphique ». Un « sens graphique » était soit « alphabétique », soit « numérique », soit « spécial » (ponctuation, logographes %, #, @, etc.). Les « alphabétiques » étaient définis comme des « lettres dans l'alphabet d'un pays » [Mackenzie, 1980, p. 16]. Cette approche naïve et alphabétocentrique était due à l'utilisation très restreinte d'ordinateurs à l'époque en dehors des États-Unis.

Unicode étant un descendant (en réalité, une extension) de ASCII, il a hérité du terme « caractère » et a introduit une panacée de termes techniques additionnels, dont nous allons considérer quelques-uns sous un prisme linguistique.

Sans doute pour éviter un conflit avec la norme ASCII ancestrale, le terme « caractère » n'est jamais utilisé sans qualification dans la spécification d'Unicode. On y trouve quatre spécialisations du terme : « caractère abstrait », « caractère codé », « caractère à éviter » (*deprecated character*) et « non-caractère »[7].

Selon la spécification Unicode, un *caractère abstrait* est défini comme

> une unité d'information utilisée pour l'organisation, le contrôle ou la représentation de données textuelles. [The Unicode Consortium, 2020, D7 in §3.4]

Nous ne pouvons nous empêcher de noter que cette définition considère la notion de « données textuelles » et son périmètre pour acquis. Or, le terme « texte » est polysémique et sa définition dépend du contexte disciplinaire. Dans le *Cambridge Dictionary of Linguistics* [Brown et Miller, 2013], le terme « texte » est défini comme suit :

> Ce terme dénotait à l'origine toute suite cohérente de phrases écrites avec une structure, typiquement marquée par plusieurs dispositifs de cohésion. Il a été étendu pour couvrir des étendues cohérentes de parole.[8]

Une définition plus générale du terme « texte » est donnée dans Wikipédia :

> En théorie littéraire, un *texte* est un objet qui peut être « lu », que cet objet soit une œuvre littéraire, un panneau de signalisation, un arrangement de bâtiments dans un pavé de maisons, ou des styles vestimentaires. C'est un ensemble cohérent de signes qui transmet un certain type de message informatif,

définition suivie par une référence à Lotman [1977]. Si nous appliquons cette définition aux processus `M-H` et `M-M` nous arrivons à la conclusion que le but du texte est d'être « lu », que ce soit par un humain ou par une machine. La « lecture », dans notre cas revient à :

1. détecter et identifier les unités élémentaires du texte,

[7] Nous utilisons la traduction officielle des caractères en français, cf. `http://unicode.org/terminology/term_en_fr.html`.

[8] *The term originally denoted any coherent sequence of written sentences with a structure, typically marked by various cohesive devices. It has been extended to cover coherent stretches of speech.*

2. appliquer la deuxième articulation pour extraire des morphèmes à partir des séquences d'unités élémentaires, et ensuite
3. appliquer la première articulation pour extraire du sens à partir des séquences de morphèmes.

Nous affirmons que ces trois opérations s'appliquent non seulement au processus `M-H`, mais aussi au processus `M-M`. Pour commencer, la machine est informée à travers différents mécanismes [YH, 2004a, p. 65] qu'elle est en train de « lire » des caractères Unicode (et non, par exemple, des pixels ou des données sonores). Elle est également informée sur la manière de lire ces données [YH, 2004a, p. 65–69], dans le but de les convertir en unités élémentaires de texte[9]. Dès que la machine est consciente du fait qu'elle est en train de lire des caractères Unicode, leur identification est possible à travers la notion de « caractère codé » :

> Un *caractère codé* est une correspondance entre un caractère abstrait et un codet (*code point*), [The Unicode Consortium, 2020, 3.4]

où un *codet* est défini de la manière suivante :

> Un *codet* est une valeur appartenant à l'espace de code Unicode [The Unicode Consortium, 2020, 3.4]

et l'*espace de code Unicode* est

> l'intervalle de nombres entiers $\{0, \ldots, 1\,114\,111\}$. [The Unicode Consortium, 2020, 3.4]

Autrement dit, dans le cadre du processus `M-M`, l'étape 1 du processus de « lecture », c'est-à-dire l'*identification des unités élémentaires* (appelés « caractères codés ») est *triviale* pour la machine, puisqu'ils sont représentés en mémoire machine par un nombre unique[10]. Aucun effort supplémentaire n'est requis.

Ce qui n'est pas le cas de l'humain dans le cadre du processus `M-H`, où les unités élémentaires sont des éléments distinctifs d'un système de langue et, en tant que tels, doivent être reconnus par les lecteurs, ou du moins ceux qui sont familiers du système d'écriture courant. Dans le cas du processus `M-H`, l'unité élémentaire de texte correspond à la notion linguistique de *graphème*, comme elle a été définie par Anis [1988a] et Günther [1988].

Peut-on comparer les notions de « caractère abstrait » (ou codé) et de « graphème » ?

Le Consortium Unicode évite de prendre position vis-à-vis de cette question. En effet, le terme « graphème » n'apparaît qu'une seule fois dans la spécification Unicode, dans la phrase très peu informative :

> Un caractère ne correspond pas forcément à l'idée préconçue de caractère que l'utilisateur peut en avoir et il faut éviter la confusion avec la notion de graphème.[11] [The Unicode Consortium, 2020, 3.4]

En guise de critique à cette affirmation nous constatons que :

[9] Les détails du stockage des unités élémentaires de texte au niveau de la machine, c'est-à-dire l'utilisation de bits et d'octets, n'affectent en rien l'étude linguistique d'Unicode.

[10] Anis [1988b, p. 30] va plus loin en considérant l'unité de base du texte informatique comme étant le bit et les nombres obtenus par combinaison de bits comme étant une *troisième articulation* : « Mais d'une certaine manière, le code binaire ne fait que radicaliser, pousser à l'extrême le principe sémio-linguistique postulé par Saussure : "dans la langue il n'y a que des différences", dans le code binaire une différence seulement. Pour reprendre les termes de Martinet, dans la langue la double articulation, dans le code binaire la multiple articulation.»

[11] *A character does not necessarily correspond to what a user thinks of as a "character" and should not be confused with a grapheme.*

- dans le cas du processus M-H, un caractère est *exactement* l'information qui, après passage par le moteur de rendu, permet à l'humain de reconnaître un graphème sans ambiguïté, donc il correspond fonctionnellement à un graphème ;
- dans le cadre du processus M-M, un caractère est *exactement* l'information nécessaire à la machine pour effectuer des traitements automatiques de la langue, de manière similaire aux graphèmes qui sont l'information nécessaire à l'humain pour traiter le langage naturel,

et pour ces raisons nous pouvons conclure que, d'un point de vue fonctionnel, les notions de caractère et de graphème sont très proches.

Néanmoins les instances de caractères ne sont pas toujours des instances de graphèmes. La principale divergence entre les deux notions est le fait que certaines écritures (comme la latine ou la cyrillique) sont utilisées par plus d'une langue et qu'Unicode traite *toutes* les langues *en même temps*. En guise d'exemple, selon Grzybek et Rusko [2009, p. 33], il existe un graphème <ch> dans la langue slovaque, et selon Wmffre [2008, p. 598], il existe un graphème <c'h> dans la langue bretonne—aucun d'eux n'est caractère Unicode. Cela vient du fait qu'Unicode doit se conformer à des conventions telles les codages nationaux ou les configurations de clavier et qu'il n'y a jamais eu de codage ou de clavier slovaques contenant <ch> ni de codage ou de clavier bretons contenant <c'h>.

Il existe même des caractères Unicode qui ne correspondent à des graphèmes dans *aucune* langue, tels que les caractères invisibles (espaces, césure potentielle, pictogrammes, émojis, etc.). De même il existe un certain nombre de caractères qui sont des graphèmes dans une langue mais dans une autre, utilisant la même écriture (comme, par exemple, <č>, utilisé en tchèque mais pas en allemand). Plus intéressant encore, le cas de caractères qui sont des graphèmes dans la langue A, qui ne sont pas graphèmes dans la langue B, mais qui peuvent être utilisés dans la langue B en tant qu'allographes d'autres graphèmes. Par exemple, les lettres <ڪ> et <ک> commutent en sindhi : <ڪنڌ> «oreille» ≠ <کنڌ> «sucre» ; en arabe, par contre, elles ne sont qu'allographes de l'arabe <ك> [YH, 2004a, p. 789], et donc ne commutent pas : <كتاب> ≡ <ڪتاب> (tous les deux «livre» en arabe[12]).

Fig. 1.5: Des logos français utilisant des allographes qui sont des graphèmes dans d'autres langues.

Le phénomène est plus facile à observer dans le design graphique où l'utilisation d'allographes est une méthode créative bien répandue. Dans les exemples de la fig. 1.5 on peut observer l'utilisation des allographes <İ>, <Ē>, <Ā> et <í>, à la place de <I>, <É>, <Â> et <i> ; ces allographes sont des graphèmes dans d'autres langues (le turc pour <İ>, le letton pour <Ē> and <Ā>, l'italien et l'espagnol pour <í>) qui sont accessibles au designer à travers des polices codées en Unicode.

1.3.2 Glyphes vs. graphes et formes de base

Meletis [2015, p. 117] définit les *graphes* pour la langue allemande comme suit :

[12] À ceci près que le *kaf* long a été historiquement utilisé dans les Corans ottomans du XIIIe siècle et après pour le mot *kufr* «impiété» [Atanasiu, 2003, chap. 12].

> Le graphe est l'unité la plus petite qui n'est pas divisible par des interstices (ceci vaut en particulier pour l'écriture imprimée et dans une moindre mesure pour l'écriture manuscrite), qui remplit un espace segmental unique et qui est matérialisée dans un système d'écriture alphabétique par des lettres mais aussi par des instances non-alphabétiques telles que la ponctuation ou les signes spéciaux comme les chiffres.

Cette définition peut facilement être appliquée aux écritures à unités atomiques séparées (par exemple, les alphabétiques, celles de l'Asie du Sud-Est, les sinogrammes, etc.). Dans le cas d'écritures qui lient systématiquement les éléments atomiques (arabe et syriaque, devanagari, etc.), la condition de « plus petite entité non-séparée par des interstices » ne peut être appliquée systématiquement. Pour ces dernières, dans le cas du texte imprimé, on peut se référer à la segmentation historique de l'écriture liée en caractères mobiles d'imprimerie en tant que solution (imparfaite) du problème de segmentation de l'écriture en graphes. Dans le cas de l'écriture manuscrite il n'existe pas de segmentation claire et des méthodes telles que la logique floue doivent être utilisées pour faire segmenter un énoncé écrit en graphes.

L'avantage de cette définition de graphe est qu'elle ne présuppose pas la connaissance de graphèmes et d'autres unités linguistiques supérieures : on peut prendre un texte dans une langue inconnue (tel que celui, par exemple, du manuscrit de Voynich), le segmenter en unités graphétiques élémentaires et ensuite procéder à une analyse des niveaux graphétiques supérieurs (c'est-à-dire, les séquences graphétiques 1-dimensionnelles dans l'espace linéaire, les séquences graphétiques 2-dimensionnelles dans l'espace de la page).

Concernant les *glyphes*, dans le standard Unicode il n'en existe pas de définition à proprement parler. Ce qui se rapproche le plus de la définition de glyphe est l'extrait suivant :

> Les glyphes représentent les formes que les caractères empruntent quand ils sont rendus ou affichés. [The Unicode Consortium, 2020, 2.2]

On en déduit la contrainte suivante : les glyphes doivent être rendus ou affichés de manière à ce que les caractères qu'ils représentent puissent être visuellement reconnus par les lecteurs familiers avec au moins une langue dans laquelle ces caractères sont utilisés.[13]

Donc si on se base sur cet extrait du standard Unicode, alors (a) les glyphes sont un aspect des caractères pour lequel Unicode ne prend aucune responsabilité, (b) liberté totale est accordée aux créateurs de fontes qui peuvent rendre les caractères à leur guise, (c) dans une optique darwinienne, uniquement le succès commercial d'une fonte peut déterminer le degré de légitimité de ses glyphes en tant que représentants des caractères Unicode. Cela peut sembler une exagération pour les grandes écritures (qui possèdent tout un écosystème de fontes plus ou moins similaires), mais devient un véritable problème pour les écritures peu dotées, pour lesquelles le nombre de fontes existant est très limité. Dürscheid [2018, §4] appelle Unicode le « gardien » des caractères, de manière similaire nous pouvons appeler l'industrie des fontes « gardienne » des glyphes[14]. Comme le montrent Manohar et Thottingal [2019], le choix d'une écriture (globalement, mais aussi au niveau de chaque graphe) est souvent un choix politique et la liberté accordée aux créateurs de fontes peut entraîner des abus.

Pour éviter cela, Unicode introduit une notion supplémentaire, celle de *glyphe indicatif* (*representative glyph*) :

[13] Techniquement, le rendu d'un caractère Unicode est contrôlé par le moteur de rendu, celui-ci fait appel aux fontes, et dans ces dernières il n'y a aucune restriction sur la forme qu'un glyphe peut avoir. On peut facilement (et impunément) créer une fonte dans laquelle le caractère LATIN CAPITAL LETTER A soit représenté par le glyphe $\left|\mathrm{B}\right|$.

[14] Dans le sens qu'un utilisateur Unicode qui n'a pas la compétence technique et artistique nécessaire pour créer une fonte est obligé d'utiliser les glyphes proposés par les fontes existantes.

> L'identité d'un caractère est établie par son nom de caractère et son glyphe indicatif, dans les tableaux de code.
>
> Un caractère peut avoir un domaine d'utilisation plus large que ce qu'une interprétation litérale de son nom peut pouvoir indiquer : la représentation codée, le nom et le glyphe indicatif doivent être évalués en contexte quand on établit l'identité d'un caractère. Par exemple, FULL STOP peut représenter un point de fin de phrase, un point d'abréviation, un séparateur décimal en anglais, un séparateur des milliers en allemand, etc. Le nom de caractère en soi est unique, mais il peut induire en erreur.
>
> La cohérence avec le glyphe indicatif ne nécessite pas que les images soient identiques ou même graphiquement similaires, cela signifie plutôt que les deux images doivent être *généralement reconnaissables en tant que représentations du même caractère*. Représenter le caractère LATIN SMALL LETTER A par le glyphe « X » serait une *violation d'identité* du caractère. [The Unicode Consortium, 2020, §3.3] (l'italique introduit par nous)

Les glyphes indicatifs de tous les caractères Unicode (non invisibles) sont proposés dans les tableaux de code Unicode[15]. La notion de glyphe indicatif est très intéressante parce que

1. elle révèle l'insuffisance de la description intensionnelle des caractères ;
2. elle induit une définition opérationnelle des glyphes : *un glyphe est une forme qui est généralement reconnue en tant que représentation d'un caractère*. Cette définition implique toujours la présence de caractères Unicode mais s'affranchit de moteur de rendu ;
3. elle montre que la relation entre caractères et glyphes relève de la sociolinguistique : un glyphe représente un caractère donné si et seulement s'il existe une communauté de personnes qui le reconnaissent comme tel.

Les notions de *graphe* en linguistique et de *glyphe* en informatique (et, en particulier, dans le contexte d'Unicode) peuvent sembler équivalentes, mais la manière dont elles sont définies rend la comparaison difficile : les *graphes* sont définis en tant qu'unités d'un système linguistique, alors que les *glyphes* dont définis en tant que rendus socialement reconnaissables de caractères.

Il est intéressant de noter que dans Unicode les sinogrammes possèdent jusqu'à six glyphes indicatifs différents, correspondant aux graphes utilisés en Chine, Hong-Kong, Taïwan, Japon, Corée et Viêt Nam. Par exemple, le caractère 伶 de codet $4F36_{16}$ (« intelligent », « acteur ») est présenté de la manière suivante dans le tableau de code :

On y distingue trois familles de formes : (a) les 1er et 6e graphes (Chine et Viêt Nam) possèdent un trait *diǎn* qui a une forme de goutte |丶| sous le radical « homme » |人| ; (b) les 2e et 3e graphes (Hong-Kong et Taïwan) ont un trait *héng* horizontal |一| sous le |人| ; (c) les 4e et 5e graphes (Japon et Corée) on une composante différente en bas à droite, comportant un trait *héng-zhé-gōu* |㇆| et un trait *shù* |丨| ([YH, 2004a, 149–150] et [Myers, 2019, 13–14]).

Nous affirmons que ces trois graphes appartiennent à des formes de base différentes, au sens de Rezec [2009]. Comme la différence des formes provient de l'utilisation de traits fondamentaux différents, les graphes garderont ces différences quelque soit leur réalisation graphique : les formes de base forment, en quelque sorte, des clusters de l'ensemble de toutes les réalisations possibles. Il n'y aura jamais de cas « intermédiaires » puisque les traits fondamentaux qui sont à la base des caractères sinographiques doivent demeurer reconnaissables en tant qu'unités distinctives du système.

[15] `https://www.unicode.org/charts/`.

1.3.3 Catégories générales de caractère vs. classes de graphèmes

Parlons maintenant de classification. Dans la classification traditionnelle des graphèmes en *logogrammes* et *phonogrammes*, ces derniers sont définis à travers leur relation à la parole. Dürscheid [2016, p. 74] définit les phonogrammes comme suit :

> Les *phonogrammes* sont des signes qui se réfèrent exclusivement au plan phonétique du système de langue.

Cette définition contredit celle de l'approche autonomiste d'Anis [1988a] qui considère l'écriture en dehors de toute interaction avec la parole. Anis propose trois classes de graphèmes : les *alphagrammes*, les *topogrammes* et les *logogrammes*. Sa définition d'un *alphagramme* est la suivante :

> ces unités distinctives, dénuées de sens par elles-mêmes, sont les composantes des unités significatives. Comme les phonèmes, les alphagrammes relèvent de la *seconde articulation*.

Un *topogramme* [Anis, 1988a, p. 116] est essentiellement un signe de ponctuation : les topogrammes contribuent à la structure et à la segmentation de séquences d'alphagrammes et de logogrammes. Les *logogrammes*, enfin, sont des unités globales ayant un signifié [Anis, 1988a, p. 139][16].

Des exemples caractéristiques d'alphagrammes sont les membres d'alphabets, d'abjads, d'abugidas, de syllabaires. Des exemples caractéristiques de logogrammes sont des graphèmes tels que <&>, <§>, les symboles monétaires <$>, <€>, etc., les symboles mathématiques <5>, <∇>, etc., les symboles d'usage général <♲>, <♂>, <☭>, etc.

Unicode propose une classification des caractères sous forme de *propriété normative* de caractère[17], appelée *catégorie générale* [The Unicode Consortium, 2020, §4.5]. Cette classification est très différente de la classification linguistique des graphèmes :

1. tous les alphagrammes appartiennent à la catégorie générale « L » (« lettre »), avec les sous-catégories suivantes : « Lu » (« lettre majuscule »), « Ll » (« lettre en bas de casse »), « Lt » (« lettre en casse de titre »), « Lm » (« modificateur ») ou « Lo » (« autre »). La catégorie générale « L » est la plus peuplée d'Unicode : elle contient 89,84% de l'ensemble total de caractères. Parmi eux, 96,13% sont monocaméraux et appartiennent à la sous-catégorie « Lo » (alphabets monocaméraux, abjads, abugidas, syllabaires, et surtout tous les caractères sinographiques) ;
2. plusieurs logogrammes[18] tels que <&>, <@>, <%>, etc., sont de catégorie générale « Po » (« autre ponctuation »), en contradiction flagrante avec leur classification linguistique ;
3. dans le cas des symboles mathématiques, Unicode utilise deux catégories générales : « N* » pour les nombres, et « Sm » pour les autres symboles mathématiques, tels que <+>, <≤>, etc. La catégorie générale « N* » comporte kes sous-catégories « Nd » (chiffres décimaux), « No » (fractions, nombres supérieurs à 9, nombres encerclés ou mis entre parenthsèes) et « Nl » (chiffres romains, hangzhou, bamum, grecs acrophoniques, gothiques, persans anciens et cunéiformes) ;

[16] Anis ne fait pas de distinction entre les fonctions sémiotiques iconique et indexique et, de ce fait, considère les pictogrammes comme un cas particulier des logogrammes. Nous n'allons pas adopter ce choix et nous allons considérer les pictogrammes comme étant distincts des logogrammes, même si la frontière entre les deux peut parfois être floue.

[17] Une propriété est « normative » dans le sens que toute application logicielle qui se proclame Unicode-compatible doit la respecter.

[18] Autres que les caractères sinographiques qui ne sont pas des purs logogrammes puisqu'ils ont une sémanticité et une phonéticité variables, cf. [YH, 2013]. Comme mentionné ci-dessus, les caractères sinographiques appartiennent à la catégorie « L ».

4. la catégorie générale « So » (« symbole, autre ») est un fourre-tout. Elle incluet des symboles tels que <©>, <°>, <☮>, mais aussi des émojis, des signes musicaux, des symboles de dessin industriel, des signes encerclés ou mis entre parenthèses, des motifs Braille, des radicaux chinois, des traits fondamentaux chinois, des hexagrammes Yi King, les signes du disque de Phaistos, des signes de notation de gestes de langue des signes, des symboles alchimiques, et pleins d'autres signes, parmi lesquels le caractère celère et impressionnant ARABIC LIGATURE BISMILLAH AR-RAHMAN ARRAHEEM

qui représente une phrase toute entière : « Au nom de Dieu clément et miséricordieux », connue sous le nom de « basmala » en français. La catégorie générale « So » représente 5% de la totalité des caractères.

La classification linguistique des graphèmes et la classification Unicode des caractères diffèrent dans leurs finalités :

- la première se focalise sur le manière dont les graphèmes contribuent à la production de sens : les alphagrammes relèvent de la deuxième articulation et donc le sens émerge de leur concaténation ; les topogrammes structurent les séquences de graphèmes et agissent donc au niveau syntaxique ; les logogrammes représentent des morphèmes ;
- la deuxième se focalise sur la manière dont les caractères sont utilisés par les applications logicielles : les caractères servant aux processus linguistiques (« L ») sont distingués des caractères de ponctuation, des symboles mathématiques et des symboles en général. Les catégories générales sont utilisées dans des textes normatifs tels que l'annexe technique sur la segmentation de texte [Davis, 2019a], qui définit les frontières du « mot » et de la « phrase » ou l'annexe de standard sur la coupure de ligne [Heninger, 2019], qui donne des spécifications pour les algorithmes de coupure de ligne. Dans les deux cas, les règles (de segmentation de texte ou de coupure de ligne) ne font impliquer que des catégories générales de caractère, et de ce fait sont universelles (indépendantes de langue, de système d'écriture, de direction d'écriture, etc.).

1.3.4 Chaînes de caractères vs. séquences de graphèmes

Un texte ne consiste quasiment jamais en un seul graphème[19]. Le plus souvent les textes en comportent plus d'un, formant des *séquences de graphèmes*. Contrairement aux données phonémiques qui ont une structure linéaire due à la structure des organes de parole humains, les séquences de graphèmes sont le plus souvent matérialisées sur des surfaces de dimension 2. L'ordre linéaire des phonèmes induit un ordre des graphèmes majoritairement linéaire, tant qu'on reste sur la même ligne. Il existe des exceptions, un bel exemple étant celui d'écriture khmer : la séquence de graphèmes représentant la séquence phonémique /kk/ est <កុ> et si on ajoute un graphème supplémentaire représentant le phonème /r/, la séquence /kkr/ s'écrit <ក្រ> (le graphéme <◌ើ> se place à gauche des deux précédents, alors qu'il s'agit d'une écriture dont la direction principale est de gauche à droite), si en plus on ajoute une voyelle /ɨə/ pour obtenir /kkrɨə/, celle-ci va encercler la séquence de graphèmes : <ើក្រឿ> (exemple tiré de [YH, 1994e]).

[19]Comme toujours il y a des exceptions à cette règle, comme le titre de l'ouvrage japonais **心** de **稲盛和夫** publié en 2019, ou l'ouvrage *S* de J.J. Abrams et D. Dorst, publié en 2013.

En linguistique, les séquences de graphèmes ont été étudiées, entre autres, par Sproat [2000], dans le cadre de la théorie de phonologie générative introduite par Chomsky et Halle [1968]. Dans cette théorie on admet l'existence de deux niveaux de représentation de données phonologiques : la *forme sous-jacente* et la *forme de surface* (avec la possibilité d'un nombre quelconque de niveaux intermédiaires). La forme de surface est obtenue en appliquant séquentiellement des règles de transformation phonologique aux données de la forme sous-jacente (chaque niveau intermédiaire étant la sortie d'une règle de transformation et l'entrée de la suivante). Une suite d'applications de règles allant de la forme sous-jacente à la forme de surface est appelée une *dérivation*. Sproat [2000] affirme que les graphèmes peuvent être obtenus en utilisant des dérivations à partir de la même forme sous-jacente que les phonèmes, autrement dit que la représentation graphémique de surface peut être obtenue en appliquant des règles de transformation aux unités de la forme sous-jacente. D'autre part, Sproat affirme que ce type de dérivation est une *relation régulière* au sens des transducteurs de type fini [Kaplan et Kay, 1994], et qu'elle est la même pour la totalité du vocabulaire d'une langue donnée.

Les relations régulières sont sans contexte, donc si γ est une dérivation et $a \cdot b$ est la concaténation de deux unités de représentation sous-jacentes, alors $\gamma(a \cdot b) = \gamma(a) \cdot \gamma(b)$. En réalité, selon Sproat [2000], nous avons non pas un seul, mais cinq *opérateurs de concaténation*, c'est-à-dire $\overset{\rightarrow}{\cdot}$, $\overset{\leftarrow}{\cdot}$, $\overset{\downarrow}{\cdot}$, $\overset{\uparrow}{\cdot}$, et $\odot$, représentant le positionnement du deuxième graphème à droite, à gauche, en dessous de, au-dessus de, ou entourant le premier graphème.

Par exemple, les règles de dérivation de l'écriture syllabique coréenne hangoul sont les suivantes [Sproat, 2000, p. 43] :

1. si σ_1 et σ_2 sont des syllabes, $\gamma(\sigma_1 \cdot \sigma_2) := \gamma(\sigma_1) \overset{\rightarrow}{\cdot} \gamma(\sigma_2)$;
2. pour l'attaque-noyau $\omega\nu$ et la coda κ, $\gamma(\omega\nu \cdot \kappa) := \gamma(\omega\nu) \overset{\downarrow}{\cdot} \gamma(\kappa)$;
3. quand la coda κ est complexe : $\kappa = \kappa_1 \cdot \kappa_2$, alors $\gamma(\kappa_1 \cdot \kappa_2) := \gamma(\kappa_1) \overset{\rightarrow}{\cdot} \gamma(\kappa_2)$;
4. pour une attaque ω et un noyau ν, either
   (a) $\gamma(\omega \cdot \nu) := \gamma(\omega) \overset{\rightarrow}{\cdot} \gamma(\nu)$, si ν appartient à l'ensemble de jamos verticaux, ou
   (b) $\gamma(\omega) \overset{\downarrow}{\cdot} \gamma(\nu)$, si ν appartient à l'ensemble de jamo horizontaux ;
5. (règle ajoutée par nous) quand le noyau ν est complexe : $\nu = \nu_1 \cdot \nu_2$, où ν_1 est horizontal et ν_2 est vertical, alors nous appliquons d'abord la règle 4(a) à $\omega\nu_1 \cdot \nu_2$ et ensuite la règle 4(b) à $\omega \cdot \nu_1$.

En guise d'illustration, appliquons ces règles à la syllabe coréenne <괽> : elle consiste en une attaque <ㄱ>, un noyau contenant deux jamo <ㅗ> et <ㅣ> le premier desquels est horizontal et le deuxième vertical, et d'une coda consistant elle aussi en deux jamo <ㄹ> et <ㅎ>. Selon la règle 5, nous appliquons d'abord la règle 4(a) à [<ㄱㅗ>]·<ㅣ> pour obtenir [[<ㄱㅗ>] $\overset{\rightarrow}{\cdot}$ <ㅣ>] et ensuite la règle 4(b) à <ㄱ>·<ㅗ>, pour obtenir [[<ㄱ> $\overset{\downarrow}{\cdot}$ <ㅗ>] $\overset{\rightarrow}{\cdot}$ <ㅣ>]. Ensuite nous appliquons la règle 3 à la coda <ㄹ>·<ㅎ> pour obtenir [<ㄹ> $\overset{\rightarrow}{\cdot}$ <ㅎ>], et enfin la règle 2 pour réunir la paire attaque-noyau et la coda, afin d'obtenir

$$[[<ㄱ> \overset{\downarrow}{\cdot} <ㅗ>] \overset{\rightarrow}{\cdot} <ㅣ>] \overset{\downarrow}{\cdot} [<ㄹ> \overset{\rightarrow}{\cdot} <ㅎ>],$$

en tant que décomposition de <괽>. Sproat appelle ce type de grammaire formelle, une *grammaire planaire*.

Parmi les nombreuses applications de la grammaire planaire de Sproat il y a aussi la diacritisation : le graphème <â> peut être représenté par le formalisme <a> $\overset{\uparrow}{\cdot}$ <^>.

Il est intéressant de noter que les opérateurs de concaténation planaires peuvent être appliqués à tous les niveaux graphiques : dans un graphème, entre graphème et signe diacritique, entre graphèmes

pour produire des séquences et former des morphèmes, entre morphèmes pour former des lignes de texte, entre lignes de texte pour former des paragraphes et des pages, entre pages pour former des volumes, de manière similaire au modèle graphétique de Meletis [2015].

Dans Unicode, il y a deux notions correspondant à la notion linguistique de séquence de graphèmes :

1. les *suites de caractères combinatoires* qui comportent un caractère de base et un ou plusieurs signes diacritiques, et
2. les *chaînes de caractères*, qui comportent plus d'un caractère de base.

Dans le premier cas, on se sert de l'opération de *combinaison* : un caractère, qui doit être de catégorie autre que « M » (« marque combinatoire ») et qu'on appelle *caractère de base*, est suivi d'un ou plusieurs *caractères combinatoires*, qui doivent être de catégorie « M ». Par exemple, pour obtenir le rendu │â│ on peut combiner le caractère de base LATIN LETTER A avec le caractère combinatoire COMBINING CIRCUMFLEX ACCENT. Cela implique que toute application logicielle Unicode-compatible doit rendre cette suite de caractères par un glyphe d'accent circonflexe placé au-dessus d'un glyphe de lettre a.

La combinaison est une opération très puissante puisqu'on peut combiner tout caractère de catégorie autre que « M » (il y a 142 142 tels caractères dans Unicode 12) avec un sous-ensemble quelconque des 2 295 caractères combinatoires, pris dans un ordre quelconque — cela résulte en un nombre astronomique de combinaisons possibles ($1{,}418 \times 10^{15}$, si on se restreint à un maximum de trois caractères combinatoires). Ce résultat est démesuré puisqu'il comptabilise également les combinaisons entre signes diacritiques d'écritures différentes, ce qui est assez rare. Rare, mais pas impossible, comme en témoignent les deux exemples suivants : (a) le logo d'une chaîne de cafés japonais appelée « Saint-Marc Café » < サンマルクカフェ > dont le dernier kana (plus petit, car modificateur de la syllabe précédente) porte un accent aigu, de la même manière que le « é » du mot français « Café » dont il est la transcription : l'accent aigu français se trouve transplanté dans le syllabaire kana ; (b) un hamza placé au-dessus du tronc vertical d'une lettre « a » (le « a » est phonétiquement équivalent à la lettre arabe *alif* qui fait partie des porteurs du *hamza*, le tronc vertical du │a│ est réminiscent de la lettre *alif*), tiré d'un T-shirt de musicien.

Les caractères combinatoires ne sont pas tous placés à la même position relative au caractère de base, et il existe 54 classes de caractères combinatoires vis-à-vis du positionnement relatif, de telles classes sont « Above » comme dans <â>, ou « Kana_voicing » comme dans <ポ>, etc. [YH, 2004a, p. 124].

Le rendu des suites de caractères combinatoires est sous la responsabilité du moteur de rendu, qui utilise des informations contenues dans les fontes, en particulier des points d'attache qui permettent de lier le caractère combinatoire au caractère de base ou les caractères combinatoires entre eux en indiquant les vecteurs de translation à effectuer [YH, 2004a, p. 755–761].

Le second cas de séquencement de caractères est celui de *chaîne de caractères*. Une chaîne de caractères est une suite de caractères dont les représentations en mémoire machine sont concomitantes. L'ordre qui doit être appliqué à ces caractères pour que le processus `M-H` aboutisse correctement est appelé *ordre logique*. Selon [The Unicode Consortium, 2020, §2.2],

> L'ordre dans lequel le texte Unicode est stocké dans la représentation de la mémoire machine est appelé *ordre logique*. Cet ordre correspond approximativement (*roughly*) à l'ordre dans lequel le texte est tapé sur un clavier ; il correspond aussi approximativement à l'ordre phonétique.

Comme sous-entendu par l'adverbe *roughly*, cette définition admet des exceptions, la plus connue étant celle des écritures thaï et lao : pour représenter la syllabe /ke:/ en khmer, l'ordre logique coïncide avec l'ordre phonétique et on place le caractère KHMER LETTER KA <ក> devant le caractère KMHER VOWEL SIGN E <ើ>, alors que le graphe du deuxième se trouve devant le graphe du premier : <ើក> ; pour obtenir la séquence graphémique correspondante en thaï ou en lao, l'ordre logique consiste à placer la voyelle THAI CHARACTER SARA E <เ> (resp. LAO VOWEL SIGN E <ເ>) *devant* la consonne THAI CHARACTER KO KAI <ก> (resp. LAO VOWEL KO <ກ>) : <เก> (resp. <ເກ>) et non pas *après* la consonne comme en khmer. Autrement dit, l'ordre logique coïncide avec l'ordre phonétique en khmer, mais pas en thaï ou en lao, ce qui est surprenant puisque les trois écritures sont très intimement liées. La raison (affichée) pour cette incongruité d'Unicode est la compatibilité avec des codages informatiques pré-existants en thaï et en lao, ainsi que les pratiques de saisie sur machine à écrire [YH, 2004a, p. 104–105].

Le grand avantage de l'« ordre logique » d'Unicode est le fait qu'il résout, du moins en mémoire machine, le problème de la mixité de directions d'écriture gauche-à-droite et droite-à-gauche, problème qui survient lorsqu'on affiche des textes mixtes français/arabe, anglais/hébreu, etc. [YH, 2004a, p. 130–133]. En mémoire machine, l'écriture latine aussi bien que l'écriture arabe sont stockées par ordre phonétique. Pour résoudre (du moins partiellement) le difficile problème d'affichage de texte mixte, Unicode attache à chaque caractère une direction par défaut : gauche-à-droite pour l'écriture latine (même si Léonard da Vinci s'amusait à écrire de droite à gauche) et droite-à-gauche pour les écritures arabe, hébraïque, syriaque et thaana. L'*algorithme bidirectionnel* [Davis, 2019b] fournit l'ordre d'affichage de chaque glyphe du rendu d'une chaîne de caractères, en se basant entre autres sur la directionnalité intrinsèque du premier caractère de la chaîne. De par l'imbrication des phrases de directions différentes et l'existence de caractères directionnellement neutres (par exemple les signes de ponctuation symétriques comme le point ou le point d'exclamation) [YH, 2004a, p. 133–141] le résultat de l'algorithme bidirectionnel peut être erroné. Pour pallier ce problème, Unicode propose des caractères tels que RIGHT-TO-LEFT EMBEDDING et POP DIRECTIONAL FORMATTING, pour forcer le comportement directionnel de certaines sous-chaînes et modifier le résultat de l'algorithme. En voici un exemple : dans la phrase ⁦A-t-il dit « Bienvenue » ?⁦ le point d'interrogation est placé en dehors des guillemets de ⁦« Bienvenue »⁦ puisqu'il appartient à la phrase <A-t-il dit…> et non pas à la phrase entre guillemets. En traduisant ⁦« Bienvenue »⁦ en hébreu, on obtient :

⁦A-t-il dit ? ברוך הבא⁦,

où le point d'interrogation est placé à gauche de la phrase citée alors qu'il aurait dû être placé à droite de celle-ci. On évite cela en plaçant un caractère LEFT-TO-RIGHT MARK devant le point d'interrogation, ce qui donne le rendu suivant :

⁦A-t-il dit ברוך הבא ?⁦.

On peut conclure que

1. l'approche formelle de Sproat [2000] est similaire aux notions de suites de caractères combinatoires et de chaînes de caractères d'Unicode, mais ne possède pas la puissance expressive des 54 classes combinatoires d'Unicode,

2. l'ordre logique d'Unicode est bien pratique pour stocker des textes directionnellement mixtes en mémoire, mais souffre d'incongruités dues à la compatibilité avec les codages pré-existants, et
3. l'algorithme bidirectionnel présente une solution pour le rendu de textes directionnellement mixtes, mais des difficulté tant linguistiques (imbrication de phrases) que techniques (existence de signes de ponctuation neutres) rend souvent nécessaire l'utilisation de caractères de « forçage de direction ».

Parenthèse : deux cas de diacritisation dynamique

Pour montrer les insuffisances tant du langage planaire de Sproat [2000] que des classes combinatoires d'Unicode, voici deux cas paradigmatiques de diacritisation dynamique.

Le premier [YH, 1994b] est celui de l'écriture hébraïque avec signes masorétiques (signes de chant pour le texte biblique). Le problème est le suivant : l'écriture hébraïque de la Bible est une abjad avec onze voyelles courtes ajoutées sous forme de diacritiques ainsi que quelques 32 signes de cantillation, appelés signes masorétiques. La distribution de combinaisons de caractère (C), de voyelle courte (V) et de signe masorétique est la suivante (nous avons effectué les calculs sur les fichiers de la *Biblia Hebraica Stuttgartensia*) :

C	∅	+V	+VV	+VVV
∅	382 084	574 548	177	0
+M	53 241	194 485	293	0
+MM	44	277	0	0
+MMM	0	0	0	0

(où les ensembles sont disjoints, c'est-à-dire que parmi les CV on ne compte pas les CVM). Au total, le corpus comporte 1 205 149 caractères, 771 255 voyelles courtes et 248 878 signes masorétiques (nous n'avons pas compté le dagesh et les points de distinction entre *sin* et *shin*). Voici un exemple d'extrait biblique où nous avons placé les voyelles courtes (ou l'absence de voyelle, *sheva*) sur fond rouge et les signes masorétiques sur fond jaune :

Pour placer les diacritiques dans le texte biblique, nous avons décrit dans [YH, 1994b] un arbre de décision avec trois issues possibles (cf. fig. 1.6) : dans le cas idéal le signe diacritique primaire est centré sur l'axe du premier caractère et le diacritique secondaire est placé à sa gauche ; si cela n'est pas possible (c'est-à-dire si l'une des deux diacritiques touche l'axe du caractère suivant) on considère les deux comme une seule entité et on la centre ; et si cela n'est toujours pas possible, on introduit du crénage entre les deux caractères. Dans l'exemple donné ci-dessus, le premier cas correspond au ④, le deuxième cas au ② et au ③ et le troisiéme cas au ①.

On peut donc en conclure que le positionnement des signes diacritiques dépend du contexte et de la nature des signes et qu'il est impossible de dire *a priori* si un signe diacritique sera centré par rapport à un caractère de base ou non.

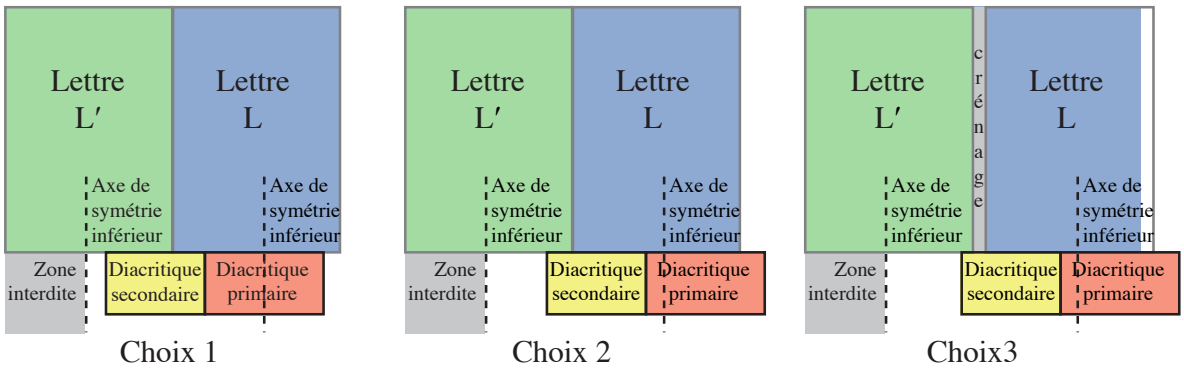


FIG. 1.6: Trois issues possibles pour le positionnement d'une voyelle et d'un signe masorétique.

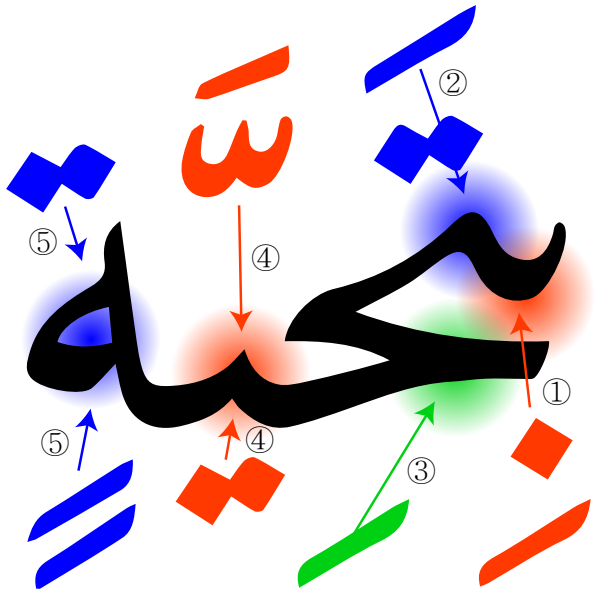


FIG. 1.7: Système d'équilibre pour les points et *harakat* d'un mot arabe.

Autre exemple [YH, 1992e, 1994c, 2006a] : celui des points, voyelles courtes et autres signes diacritiques (*harakat*) de l'arabe. Sur la fig. 1.7 on peut voir un mot (بِتَحِيَّةٍ «avec une salutation» extrait du Coran 4:86) avec les points et les *harakat* correspondant aux cinq graphèmes (auxquels nous avons associé les couleurs rouge, bleu, vert, rouge et bleu). Chaque graphe a un centre qui fonctionne comme point d'attraction des points et *harakat* qui lui correspondent et comme point de répulsion envers ceux des autres graphes. Nous avons représenté les forces d'attraction par des flèches. À ces forces d'attraction/répulsion s'ajoute une contrainte : les «parties mobiles» (points, *harakat*) doivent rester à une distance suffisante des «parties fixes» (lettres de base) pour des raisons de lisibilité (et cette distance dépend, bien sûr, du corps).

Chaque mot arabe est donc un *système d'équilibre de trois types de forces*. Sur la fig. 1.7 on peut constater que la force d'attraction ① est presque verticale alors que ② et ③ sont de plus en plus obliques. Le *kasra* vert du ③ ne peut se déplacer plus à droite car la force de répulsion du *tah* (bleu) l'en empêche, pour éviter que le lecteur l'associe à cette lettre. Le point du *bah* rouge ① se trouve sous le *hah* vert ③ pour des raisons esthétiques mais ne doit, en aucun cas, être considéré comme

un point du *hah* qui deviendrait alors un *djim*, d'où son placement très à droite du centre d'attraction vert. Pour les deux dernières lettres il n'y a aucune confusion et les signes sont placés quasiment sur les axes verticaux des graphes (forces ④ et ⑤).

La lecture de l'arabe (ainsi que de tout autre système d'écriture de type abjad) demande un effort cognitif de détection de schème : une fois la racine reconnue, le lecteur doit trouver le schème qui convient selon la la position du mot dans la phrase. Il s'agit d'une analyse morphologique et, dans le cas de texte non-voyellé, elle repose sur une combinatoire mentale de voyelles courtes pour chaque candidat de racine et de schème. Comme nous venons de le constater, cette opération cognitive de désambiguïsation est accompagnée d'une autre opération de plus bas niveau : pendant la lecture, le lecteur doit d'abord établir la correspondance entre les « parties fixes » et les « parties mobiles » (points et *harakat*, quand ceux-ci sont indiqués). Cette opération est triviale pour les fontes à ligne de base horizontale et sans ligatures esthétiques comme celle-ci :

où les « parties fixes » et les « parties mobiles » correspondantes forment des blocs verticaux bien distincts, mais demande des effors cognitifs accrus lorsque l'écriture devient de plus en plus calligraphique.

1.4 Ligatures

Dans la majorité des cas où une chaîne de caractères est fournie en entrée à un moteur de rendu dans le cadre du processus `M-H`, le principe de régularité de Sproat s'applique. Il s'ensuit que *la dérivation de la chaîne d'unités linguistiques adjacentes est la concaténation planaire des dérivations des unités linguistiques*. Les exceptions à cette règle sont appelées *ligatures* [YH, 1995c].

Les *ligatures* sont des graphes obtenus par la fusion de graphes adjacents. Elles peuvent être *optionnelles* (dans le sens que les graphes adjacents peuvent aussi, sous certaines conditions, ne pas fusionner pour former une ligature) ou *obligatoires*. L'utilisation de ligatures peut, ou non, avoir un impact sur l'analyse linguistique.

- Par définition, les *ligatures obligatoires* sont celles qui ont lieu systématiquement lorsque les graphèmes sous-jacents sont adjacents. Étant obligatoires, Unicode a choisi de ne pas les coder et de reléguer leur gestion au moteur de rendu. L'exemple le plus célèbre est celui de la ligature arabe *lam-alif* |لا| (à comparer avec l'hypothétique non-ligaturé |ﻠﺍ|). L'utilisation de la ligature *lam-alif* est une règle fondamentale de l'écriture arabe, comme dans la phrase suivante, composée en *Mashq Kufi* non-ponctué du XI[e] siècle [Mousavi Jazayeri et al., 2017] :

(لا رأيتُ ولا سَمعتُ, « Je n'ai ni vu, ni entendu »). La ligature *lam-alif* est utilisée dans *toutes* les langues d'écriture arabe. Elle est enseignée dans les écoles en tant qu'élément constituant de l'alphabet arabe [Dichy, 2019] et les machines à écrire arabes possèdent une touche dédiée. Néanmoins, malgré sa présence universelle dans le monde arabe, le *lam-alif* reste une ligature et de ce fait, Unicode ne propose pas de caractère pour la coder[20].

[20] Ceci n'est qu'en partie vrai. Il existe un caractère de type « forme de présentation » qui code cette ligature : ARABIC LIGATURE

- Les *ligatures optionnelles* sont celles qui apparaissent sous certaines conditions quand deux graphèmes sont adjacents.

On subdivise les ligatures optionnelles en deux classes : les *ligatures esthétiques* et les *ligatures linguistiquement motivées* [YH, 1995c] :

- Les *ligatures esthétiques* contribuent uniquement à la lisibilité et la qualité esthétique du texte écrit. Des exemples caractéristiques en sont le latin │fi│, l'arabe │ﺞﻣ│ et l'arménien │մվ│ (à comparer avec les non-ligaturés │fi│, │ﻣﺞ│ and │մվ│). La raison d'être des ligatures esthétiques est purement visuelle : pour éviter la superposition de la goutte du │f│ et du point du │i│ dans le cas du │fi│, pour compresser le texte en écrivant │ﺞﻣ│ verticalement, pour éviter le blanc excessif entre les graphes dansle cas du │մվ│.
  Notons que même si les ligatures esthétiques n'ont pas de motivation linguistique, leur activation peut dépendre de la langue courante. Par exemple, le turc n'utilise pas les ligatures │fi│ et │ffi│ pour éviter l'ambiguïté entre les graphèmes <i> et <ı> (sans point), puisque la fusion avec le graph f invaliderait une composante du système graphémique turc.
- Les *ligatures linguistiquement motivées* ont un statut intermédiaire entre celui de graphème (individuel) et celui de séquence de graphèmes. Des exemples typiques sont le français <œ> et le néerlandais <ij>. Tous deux se comportent comme des graphèmes individuels lors d'un changement de casse : <Œttingen>, <IJmegen> (et non pas *│Oettingen│ ou *│Ijmegen│), ils ont des représentations phonétiques spécifiques (différentes de celles des graphèmes sous-jacents). Contrairement à la ligature arabe *lam-alif*, ils n'apparaissent pas dans les claviers de machine à écrire[21]. Ils peuvent être qualifiés de « citoyens de second ordre » du système graphémique du français et du néerlandais : ils n'apparaissent pas dans les grammaires scolaires et sont relativement difficiles à saisir sur les claviers d'ordinateur.

La frontière entre ligature esthétique et ligature linguistiquement motivée peut être floue : par exemple, en allemand, les f-ligatures sont des *marqueurs morphologiques indirects* puisqu'elles ne peuvent être utilisées qu'intramorphiquement. Dans la tradition typographique allemande, les ligatures intermorphémiques sont interdites : on écrira │Kaufleute│, │Auffassung│, etc.

Les ligatures sont intéressantes d'un point de vue théorique parce qu'elles défient la définition de la notion de système d'écriture en tant que système d'unités élémentaires distinctives permettant une double articulation. Ainsi, selon Nehrlich [2012, p. 30] (qui se réfère au système d'écriture de l'allemand) :

> La ligature met en doute les propriétés principales du système d'écriture, l'argument le plus poignant étant le fait que l'existence des ligatures rend problématique le concept de lettre. Les lettres sont les graphèmes qui constituent l'alphabet, mais cela n'est valable que tant qu'on reste au niveau de la représentation abstraite. Dès que l'écriture se matérialise, le concept de lettre perd sa validité : la présence de ligatures rend caduque la définition de la lettre comme étant ce qui est séparé d'interstices à l'intérieur d'un mot.

LAM WITH ALIF ISOLATED FORM, mais son utilisation est déconseillée par Unicode : «[Les caractères de type « forme de présentation »] sont inclus pour des raisons de compatibilité avec les codages pré-existants et les implémentations traditionnelles qui les utilisent en tant que caractères. À la place de celles-ci, il est recommandé d'utiliser des lettres du bloc arabe ». [The Unicode Consortium, 2020, §9.2]

[21] À noter néanmoins que les machines de composition Monotype/Linotype, utilisées dès la fin du XIX^e^ siècle et pendant tout le XX^e^ siècle, possédaient des touches dédiées aux graphèmes <œ> et <ij>.

En effet, d'un point de vue systémique, les ligatures (esthétiques) sont superflues puisqu'elles ne portent aucune information linguistique et, dans une perspective darwinienne, les propriétés superflues ont, normalement, tendance à disparaître dans un système qui évolue. Mais les ligatures existent aussi longtemps qu'existe l'écriture et leur disparition ne semble pas être imminente. Les ligatures nous font réaliser que, de même que la lumière a une double nature particule/onde, les graphèmes ont aussi une double nature puisqu'ils portent aussi bien de l'information graphique que de l'information linguistique. De manière similaire à l'expérience des fentes de Young, les ligatures révèlent (du moins pour les écritures occidentales) cette double nature des graphèmes.

La double nature des graphèmes (et donc aussi des caractères Unicode qui représentent les graphèmes dans le monde numérique) est la principale cause des différences entre les processus `M-H` et `M-M`, et il n'est pas surprenant que le Consortium Unicode a examiné la question des ligatures très soigneusement.

En effet, Unicode distingue très clairement les ligatures linguistiquement motivées des ligatures esthétiques et obligatoires. Les premières sont codées en tant que caractères, les deuxièmes ne le sont pas[22].

Les ligatures esthétiques sont gérées automatiquement par les moteurs de rendu en utilisant des informations provenant des fontes. L'utilisateur peut désactiver leur utilisation en insérant un caractère spécial entre les caractères ligaturés, appelé ZERO WIDTH NON-JOINER. C'est lui qui est utilisé, par exemple, en allemand, lorsqu'il s'agit d'éviter les ligatures intermorphémiques.

Le cas de l'arabe

Contrairement à l'écriture latine (imprimée) dont les graphes sont séparés par des interstices, dans l'écriture arabe les graphes de graphèmes adjacents interagissent, et cela selon deux niveaux :

1. au premier niveau, un graphe quadriforme[23] est nécessairement[24] lié au graphe qui le suit. Les traits de liaison sont nécessairement horizontaux et se placent sur la ligne de base, comme dans |جج| ;
2. au deuxième niveau, des ligatures esthétiques sont appliquées. Dans ce cas, les graphes sont combinés verticalement ou diagonalement, comme dans |ﺟﺞ| [YH, 1994c].

Puisqu'il existe deux niveaux distincts d'interaction entre les graphes, on doit être capable d'interférer au premier niveau (séparer deux graphes qui normalement sont liés) ou au deuxième niveau (éviter l'utilisation d'une ligature esthétique tout en gardant la liaison normale des graphes). Pour permettre cette interaction à deux niveaux, Unicode recommande l'utilisation de deux caractères invisibles :

1. le caractère ZERO WIDTH NON-JOINER (déjà mentionné) agit au premier niveau et sépare les graphes en changeant leur forme contextuelle (un graphe suivi de ce caractère va passer de la forme initiale à la forme isolée et de la forme médiane à la forme finale ; un graphe précédé de ce caractère va passer de la forme médiane à la forme initiale et de la forme finale à la forme isolée) ;

[22] Si on fait abstraction des caractères de type « forme de présentation » qui existent, mais dont l'utilisation est déconseille par le Consortium-même qui les a introduits dans le codage.

[23] Dans le système d'écriture de la langue arabe, les graphes |ا د ذ ر ز و| sont biformes(formes isolée et finale), le graphe |ء| est monoforme et tous les autres graphes sont quadriformes (formes isolée, initiale, médiane et finale). On a le même phénomène en syriaque et en mongole (qui est un dérivé du syriaque).

[24] À l'exception de reformes expérimentales de l'écriture arabe comme celles décrites dans [YH, 1998].

2. le caractère ZERO WIDTH JOINER agit au deuxième niveau en empêchant la formation de ligature esthétique mais en préservant la liaison normale des graphes et la formation de formes contextuelles.

En guise d'exemple, comparons les trois paires de graphes :

par défaut	avec ZERO WIDTH JOINER	avec ZERO WIDTH NON-JOINER
جح	جـجـ	جـ جـ

Jusqu'ici nous avons considéré la propriété de forme contextuelle comme une propriété graphétique et non-linguistique. Néanmoins, dans certains cas le non-respect des règles contextuelles peut contribuer à la production de sens en changeant la nature du graphème de phonographique à logographique. Par exemple, le graphe |هـ| (la forme initiale du graphème <ه>) est souvent utilisé en tant qu'abréviation de سنة هجرية «année hégirienne», et est donc un logogramme. Le même graphe peut avoir d'autres significations : par exemple, dans le dictionnaire franco-arabe *Mounged de poche*, un certain nombre d'abréviations se servent d'allographes en forme initiale : |مـ| pour le genre féminin (مؤمع), |جـ| pour le pluriel (جمع), |هـ| pour les pronoms à référents non-humains, et le même graphème en forme isolée |ه| pour les pronoms à référents humains. À noter que dans ce dictionnaire, aucun point d'abréviation n'est utilisé et donc la forme contextuelle est le *seul* indicateur de la nature logographique des graphèmes.

Sous Unicode, la transgression des règles contextuelles pour les graphes arabes fait partie des fonctions du caractère ZERO WIDTH NON-JOINER : pour obtenir l'abréviation en forme initiale |هـ| à travers le processus M-H, le caractère ARABIC LETTER HEH doit être suivi par le caractère ZERO WIDTH JOINER. Comme cette opération produit du sens en changeant la nature du graphème, le caractère ZERO WIDTH JOINER est également nécessaire dans le cadre du processus M-M, *même si la machine n'est pas obligée de visualiser le texte pour le traiter*.

1.5 Les textèmes

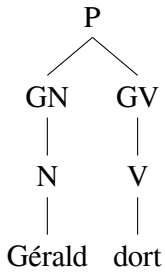
Le formalisme de description linguistique HPSG (*head-driven phrase structure grammar*) ([Pollard et Sag, 1994] et [Borsley et Börjars, 2011, Chap. 1 & 2]) permet une analyse des différentes couches linguistiques d'un texte, mises en parallèle. Il a été appliqué aux graphèmes par Sproat [2000] et nous l'avons étendu à la graphétique et aux différentes opérations des séquences graphémiques et graphétiques en introduisant la notion de *textème* [YH et Bella, 2005a ; Bella et YH, 2007], qui sera l'objet de cette section.

1.5.1 Matrices attributs-valeurs

Le formalisme HPSG fonctionne comme suit : pour chaque nœud de l'arbre syntaxique en constituants, on introduit un certain nombre d'attributs[25], dont deux obligatoires : PHON (la représentation phonétique du mot ou de la suite de mots concernés) et SYNSEM (les informations syntaxiques et sémantiques). Lorsqu'il s'agit d'un nœud intermédiaire de l'arbre syntaxique, un troisième attribut vient s'ajouter : DTRS (*daughters* = filles) dont la valeur est une liste d'attributs correspondant aux nœuds fils du nœud intermédiaire en question.

[25] Dans le cadre de ce formalisme, un attribut est une paire clé-valeur dont la valeur peut être un type de données atomique ou une liste d'autres attributs. Il s'agit donc d'un type de données récursif.

Prenons la phrase « Gérald dort », dont voici l'arbre syntaxique :



Pour représenter graphiquement les attributs imbriqués, nous allons utiliser une construction appelée AVM (matrice d'attributs et de valeurs). Pour le nœud P de l'arbre syntaxique ci-dessus, elle prend l'aspect suivant :

$$\begin{bmatrix} \textsc{phon} & /\text{ʒeʁald dɔʁ}/ \\ \textsc{synsem} & \dots \\ \textsc{dtrs} & \begin{bmatrix} \textsc{nhd} & \begin{bmatrix} \textsc{phon} & /\text{ʒeʁald}/ \\ \textsc{synsem} & \dots \end{bmatrix} \\ \textsc{hd} & \begin{bmatrix} \textsc{phon} & /\text{dɔʁ}/ \\ \textsc{synsem} & \dots \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

La « tête » de la phrase, au sens de la syntaxe des dépendances, est « dort ». C'est pour cette raison que le sous-attribut de DTRS qui correspond au premier mot (« Gérald ») est un NHD (*non-head*, une « non-tête ») et celui du deuxième mot (« dort ») est un HD (*head*, la « tête »). Nous avons également ajouté les représentations phonétiques de la phrase tout entière et des deux mots séparément.

Il ne reste plus qu'à remplir les cases SYNSEM. Les informations que nous allons y placer sont de type morphosyntaxique, c'est-à-dire qu'elles concernent le rôle syntaxique de chaque mot ainsi que sa morphologie. Au niveau syntaxique, il suffit d'indiquer dans un attribut CAT l'éventuelle dépendance syntaxique à laquelle participe le mot. Notre phrase n'a qu'une seule dépendance : de « dort » vers « Gérald », dépendance de type sujet puisque « Gérald » est le sujet de « dort ». Voici comment cette dépendance est représentée dans l'AVM :

$$\begin{bmatrix} \textsc{phon} & /\text{ʒeʁald dɔʁ}/ \\ \textsc{synsem} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \textit{verbe} \end{bmatrix} \end{bmatrix} \\ \textsc{dtrs} & \begin{bmatrix} \textsc{nhd} & \begin{bmatrix} \textsc{phon} & /\text{ʒeʁald}/ \\ \textsc{synsem} & \boxed{1} \begin{bmatrix} \textsc{cat} & \textit{nom} \\ \dots & \dots \end{bmatrix} \end{bmatrix} \\ \textsc{hd} & \begin{bmatrix} \textsc{phon} & /\text{dɔʁ}/ \\ \textsc{synsem} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \textit{verbe} \\ \textsc{val} & \begin{bmatrix} \textsc{suj} & \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix} \\ \dots & \dots \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Le lecteur remarquera que l'attribut VAL du verbe contient un sous-attribut SUJ (le type de la dépendance) dont la valeur est 1, ce chiffre encadré étant une référence vers l'attribut SYNSEM du nom « Gérald ». À l'aide de telles références internes on peut représenter tout l'arbre syntaxique de dépendances à l'intérieur-même de l'AVM.

Il ne reste plus qu'à ajouter les informations morphosyntaxiques : pour le nom ce sera le *genre*, le *nombre* (dans certaines langues aussi le *cas*) ; pour le verbe ce sera le *mode*, le *temps*, le *nombre*, la *personne*. Sproat rajoute un autre attribut : la sémantique SEM, qui fournit le lemme du mot (c'est-à-dire sa forme standard, pour les noms ce sera le nominatif singulier, pour les verbes l'infinitif) traduit dans une langue compréhensible par le lecteur (dans notre cas, ce sera le français). Ainsi nous obtenons l'AVM suivante, pour notre petite phrase « Gérald dort » :

$$\begin{bmatrix}
\text{PHON} & /\text{ʒeʁald dɔʁ}/ & & & \\
\text{SYNSEM} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{HEAD} & \textit{verbe}\end{bmatrix}\end{bmatrix} & & & \\
\text{DTRS} & \begin{bmatrix}
\text{NHD} & \begin{bmatrix}\text{PHON} & /\text{ʒeʁald}/ & & \\ \text{SYNSEM} & \boxed{1} & \begin{bmatrix}\text{CAT} & \textit{nom}\\ \text{GEN} & \textit{masc.}\\ \text{NUM} & \textit{sing.}\\ \text{SEM} & \textit{Gérald}\end{bmatrix}\end{bmatrix} & & \\
\text{HD} & \begin{bmatrix}\text{PHON} & /\text{dɔʁ}/ & & \\ \text{SYNSEM} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{HEAD} & \textit{verbe}\\ \text{val} & \begin{bmatrix}\text{SUJ} & \langle\boxed{1}\rangle\end{bmatrix}\end{bmatrix}\\ \text{MOD} & \textit{indic.}\\ \text{TEN} & \textit{prés.}\\ \text{NUM} & \textit{sing.}\\ \text{PER} & \textit{3}\\ \text{SEM} & \textit{dormir}\end{bmatrix}\end{bmatrix} & &
\end{bmatrix}
\end{bmatrix}$$

Avant de revenir à la graphématique notons que la phonétique est ici honteusement privilégiée : non seulement elle est présente au niveau de chaque nœud, mais en plus elle est placée au « premier rang », d'égale importance avec SYNSEM. Implicitement on considère donc que l'ensemble de toutes les informations qui se trouvent sous SYNSEM est d'égale importance que la simple information de représentation phonétique du nœud.

1.5.2 Intégrer la langue écrite dans HPSG

Sproat [2000] définit tout d'abord un nouvel attribut pour la représentation écrite des nœuds, et comme ce qui l'intéresse est une représentation bien définie et normalisée, il l'appelle ORTH (orthographe). Au risque de déplaire aux Saussuriens, il place cet attribut au même niveau que PHON. Ensuite il s'intéresse à la *relation entre phonèmes et graphèmes* et il rend cette relation visible au niveau de l'AVM. Il numérote donc les phonèmes de PHON et définit ORTH comme étant un ensemble

de graphèmes numérotés — c'est la numérotation qui établit la correspondance entre phonèmes et graphèmes. Ainsi, on aura, par exemple :

$$\begin{bmatrix} \text{PHON} & /ʒ_{1*}e_{2*}ʁ_{3*}a_{4*}l_{5*}d_{6*}/ \\ \text{ORTH} & \{G_1, é_2, r_3, a_4, l_5, d_6\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{nom} \\ \text{GEN} & \textit{masc.} \\ \text{NUM} & \textit{sing.} \\ \text{SEM} & \textit{Gérald} \end{bmatrix} \end{bmatrix}$$

L'astérisque ajouté aux indices des phonèmes indique que c'est le phonème qui se fait représenter par un graphème, et non l'inverse.

Il est bien connu que la correspondance entre phonèmes et graphèmes peut être complexe. Pour couvrir tous les cas possibles, Sproat se sert d'indices,de phonèmes sans indices et aussi du symbole ∅ de « phonème vide ». Ainsi l'abréviation <n°> (deux graphèmes) qui est prononcée /nymeʁo/ (six phonèmes, deux graphèmes), aura l'AVM suivante :

$$\begin{bmatrix} \text{PHON} & /n_{1*}y_{2*}m_{3*}e_{4*}ʁ_{5*}o_{6*}/ \\ \text{ORTH} & \{n_1, {}^{o}{}_6\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{nom} \\ \text{GEN} & \textit{masc.} \\ \text{NUM} & \textit{sing.} \\ \text{SEM} & \textit{numéro} \end{bmatrix} \end{bmatrix}$$

et le mot <dette> (cinq graphèmes) qui est prononcé /dɛt/ (trois phonèmes, cinq graphèmes), aura l'AVM suivant :

$$\begin{bmatrix} \text{PHON} & /d_{1*}ɛ_{2*}t_{3*}∅_{4*}∅_{5*}/ \\ \text{ORTH} & \{d_1, e_2, t_3, t_4, e_5\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{nom} \\ \text{GEN} & \textit{fém.} \\ \text{NUM} & \textit{sing.} \\ \text{SEM} & \textit{dette} \end{bmatrix} \end{bmatrix}$$

La fonction d'un graphème peut aussi être morphologique plutôt que phonétique. Ainsi, dans le mot <paris> (pluriel de <pari>) le <s> est muet mais indique le pluriel du mot[26]. On indiquera cela dans l'AVM en indexant le trait grammatical NUM concerné :

[26]Une règle qui est loin d'être universelle, il suffit de penser à des mots comme <fois>, <avis>, <sursis> qui prennent un <s> muet aussi bien au singulier qu'au pluriel.

$$\begin{bmatrix} \text{PHON} & /p_{1*}a_{2*}ʁ_{3*}i_{4*}/ \\ \text{ORTH} & \{p_1, a_2, r_3, i_4, s_5\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{nom} \\ \text{GEN} & \textit{masc.} \\ \text{NUM} & \textit{plur.}_{5*} \\ \text{SEM} & \textit{pari} \end{bmatrix} \end{bmatrix}$$

1.5.3 Les textèmes en tant que généralisation des AVM de Sproat

Dans [YH et Bella, 2005a ; Bella et YH, 2007] nous avons proposé une extension des AVM de Sproat aux graphèmes et graphes, et une implémentation de cette extension aux caractères Unicode et aux glyphes (identifiés par leurs identifiant dans des fontes). Nous appelons *textème* une AVM du type

$$\begin{bmatrix} \text{CHAR} & <a> \\ \text{autres propriétés} & \\ \text{GLYPH} & |a| \end{bmatrix}$$

qui doit au moins contenir un attribut de caractère et un attribut de glyphe. Ainsi, par exemple, un textème de lettre arabe peut contenir la forme contextuelle du graphe (ce qui est bien plus élégant que l'approche des caractères spéciaux Unicode pour forcer la forme contextuelle) et ainsi l'AVM de Sproat pour le mot arabe بيوت «maisons»

$$\begin{bmatrix} \text{PHON} & /bu_{1*}y_{2*}u:_{3*}t_{4*}/ \\ \text{ORTH} & \{ب_1, ي_2, و_3, ت_4\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{nom} \\ \text{GEN} & \textit{masc.} \\ \text{NUM} & \textit{pl.} \\ \text{CAS} & \textit{nom.} \\ \text{SEM} & \textit{maison} \end{bmatrix} \end{bmatrix}$$

devient

$$\begin{bmatrix} \text{PHON} & /bu_{1*}y_{2*}u:_{3*}t_{4*}/ \\ \text{ORTH} & \left\{ \begin{bmatrix} \text{CHAR} & <ب>_1 \\ \text{FORM} & 1 \\ \text{GLYPH} & |بـ| \end{bmatrix}_1, \begin{bmatrix} \text{CHAR} & <ي>_2 \\ \text{FORM} & 2 \\ \text{GLYPH} & |يـ| \end{bmatrix}_2, \begin{bmatrix} \text{CHAR} & <و>_3 \\ \text{FORM} & 3 \\ \text{GLYPH} & |و| \end{bmatrix}_3, \begin{bmatrix} \text{CHAR} & <ت>_3 \\ \text{FORM} & 0 \\ \text{GLYPH} & |تـ| \end{bmatrix}_4 \right\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{nom} \\ \text{GEN} & \textit{masc.} \\ \text{NUM} & \textit{pl.} \\ \text{CAS} & \textit{nom.} \\ \text{SEM} & \textit{maison} \end{bmatrix} \end{bmatrix}.$$

Les textèmes permettent l'ajout d'informations linguistiques, ainsi on peut ajouter à l'AVM ci-dessus de l'information sur la racine sémitique ROOT et sur le schème SCHEME :

$$\begin{bmatrix}
\text{PHON} & /\text{bu}_{1*}\,\text{y}_{2*}\,\text{u:}_{3*}\,\text{t}_{4*}/ \\
\text{ORTH} & \left\{ \begin{bmatrix} \text{CHAR} & <\text{ب}>_1 \\ \text{FORM} & 1 \\ \text{GLYPH} & |\text{بـ}| \end{bmatrix}_1, \begin{bmatrix} \text{CHAR} & <\text{ي}>_2 \\ \text{FORM} & 2 \\ \text{GLYPH} & |\text{ـيـ}| \end{bmatrix}_2, \begin{bmatrix} \text{CHAR} & <\text{و}>_3 \\ \text{FORM} & 3 \\ \text{GLYPH} & |\text{و}| \end{bmatrix}_3, \begin{bmatrix} \text{CHAR} & <\text{ت}>_3 \\ \text{FORM} & 0 \\ \text{GLYPH} & |\text{ت}| \end{bmatrix}_4 \right\} \\
\text{SYNSEM} & \begin{bmatrix}
\text{CAT} & \textit{nom} \\
\text{GEN} & \textit{masc.} \\
\text{NUM} & \textit{pl.} \\
\text{CAS} & \textit{nom.} \\
\text{SEM} & \textit{maison} \\
\text{ROOT} & \{\text{ب}_{1\to1}, \text{ي}_{2\to2}, \text{ت}_{3\to4}\} \\
\text{SCHEME} & \{\text{◌ُ}_{1,(1,2)}, \text{◌ُ}_{2,(2,3)}, \text{و}_{3,(2,3)}, \text{◌ْ}_{4,(3,4)}\}
\end{bmatrix}
\end{bmatrix}$$

où la notation $\{\text{ب}_{1\to1}, \text{ي}_{2\to2}, \text{ت}_{3\to4}\}$ indique qu'il s'agit de la racine *ba* (1er graphème) – *ya* (2e graphème) – *ta* (4e graphème) et la notation $\{\text{◌ُ}_{1,(1,2)}, \text{◌ُ}_{2,(2,3)}, \text{و}_{3,(2,3)}, \text{◌ْ}_{4,(3,4)}\}$ indique que la racine est combinée avec le schème *damma* (à placer entre le 1er et le 2e élément de la racine) - *damma* (à placer entre le 2e et le 3e élément de la racine) - *wa* (à placer entre le 2e et le 3e élément de la racine) - *soukoun* (à placer après le 3e élément de la racine).

Opérations sur les textèmes

Parmi les avantages des textèmes se trouve le fait qu'ils représentent aussi bien le côté graphémique (caractères) que le côté graphétique (glyphes) des unités de la modalité écrite de la langue. De ce fait ils peuvent être utilisés pour modéliser un certain nombre d'opérations appliquées aux caractères ou aux glyphes, ou aux deux en même temps.

Un premier type d'opération est celui des tables OpenType GSUB et GPOS [YH, 2004a, p. 746–785]. Il s'agit d'opérations au niveau les glyphes dans le cadre d'une fonte. En voici quelques-unes et leur modélisation à travers les textèmes :

- *substitution simple* : un glyphe est remplacé par un autre glyphe allographe et ceci de manière systématique. Exemple caractéristique, le passage en petites capitales (l'information additionnelle est stockée dans un attribut SMALLCAPS) :

$$\begin{bmatrix} \text{CHAR} & <\text{a}> \\ \text{GLYPH} & |\text{a}| \end{bmatrix} \rightarrow \begin{bmatrix} \text{CHAR} & <\text{a}> \\ \text{SMALLCAPS} & 1 \\ \text{GLYPH} & |\text{A}| \end{bmatrix};$$

- *substitution multiple* : un glyphe est remplacé par deux glyphes ou plus. Par exemple, on pourrait souhaiter remplacer le glyphe sinographique |㎑| par la séquence de glyphes |kHz| (on génère des textèmes sans caractère) :

$$\begin{bmatrix} \text{CHAR} & <\text{㎑}> \\ \text{GLYPH} & |\text{㎑}| \end{bmatrix} \rightarrow \begin{bmatrix} \text{CHAR} & <\text{㎑}> \\ \text{GLYPH} & |\text{k}| \end{bmatrix}\begin{bmatrix} \text{CHAR} & <> \\ \text{GLYPH} & |\text{H}| \end{bmatrix}\begin{bmatrix} \text{CHAR} & <> \\ \text{GLYPH} & |\text{z}| \end{bmatrix};$$

- *substitution alternative* : on remplace un glyphe par un autre glyphe allographe, parmi plusieurs (une interface homme-machine peut être nécessaire pour choisir le bon allographe) :

$$\begin{bmatrix} \text{CHAR} & \text{<A>} \\ \text{GLYPH} & \text{|A|} \end{bmatrix} \rightarrow \begin{bmatrix} \text{CHAR} & \text{<A>} \\ \text{alt=3} & \\ \text{GLYPH} & \text{|}\mathfrak{A}\text{|} \end{bmatrix},$$

  la différence avec a substitution simple étant que cette dernière produit toujours le même glyphe et n'offre pas de choix ;
- *substitution de ligature* : une ligature est formée à partir de plusieurs glyphes, cette fois on introduit des graphèmes sans glyphe :

$$\begin{bmatrix} \text{CHAR} & \text{<f>} \\ \text{GLYPH} & \text{|f|} \end{bmatrix}\begin{bmatrix} \text{CHAR} & \text{<i>} \\ \text{GLYPH} & \text{|i|} \end{bmatrix} \rightarrow \begin{bmatrix} \text{CHAR} & \text{<f>} \\ \text{GLYPH} & \text{|fi|} \end{bmatrix}\begin{bmatrix} \text{CHAR} & \text{<i>} \\ \text{GLYPH} & \text{||} \end{bmatrix};$$

- *ajustement simple* : changement du positionnement d'un glyphe. Nous introduisons des attributs DX et DY pour indiquer les décalages horizontal et vertical du glyphe (en laissant le vecteur de chasse du glyphe inchangé). Un exemple caractéristique est le logo |TEX|, qui peut être considéré une instance allographique de la chaîne de graphèmes <TEX> :

$$\begin{bmatrix} \text{CHAR} & \text{<T>} \\ \text{KERN} & -.1667\text{em} \\ \text{GLYPH} & \text{|T|} \end{bmatrix}\begin{bmatrix} \text{CHAR} & \text{<E>} \\ \text{KERN} & -.125\text{em} \\ \text{DY} & -.5\text{ex} \\ \text{GLYPH} & \text{|E|} \end{bmatrix}\begin{bmatrix} \text{CHAR} & \text{<X>} \\ \text{GLYPH} & \text{|X|} \end{bmatrix},$$

  ou comme un logogramme en soi

$$\begin{bmatrix} \text{CHAR} & \text{<TEX}_{\square}\text{>} \\ \text{KERN} & -.1667\text{em} \\ \text{GLYPH} & \text{|T|} \end{bmatrix}\begin{bmatrix} \text{CHAR} & \text{<>} \\ \text{KERN} & -.125\text{em} \\ \text{DY} & -.5\text{ex} \\ \text{GLYPH} & \text{|E|} \end{bmatrix}\begin{bmatrix} \text{CHAR} & \text{<>} \\ \text{GLYPH} & \text{|X|} \end{bmatrix},$$

  où l'attribut KERN indique le crénage entre un textème et son successeur et « em » est l'unité de cadratin ;
- *ajustement de paire* : ajustement d'un glyphe par rapport à son prédécesseur. On définit des attributs KERN et VKERN pour obtenir les coordonnées horizontale et verticale du vecteur d'ajustement systématique entre les glyphes (qui s'ajoute au vecteur de chasse du premier glyphe).

Gestion de la coupure de ligne

Un autre problème qui peut être résolu à l'aide des textèmes est celui de la coupure de séquence graphémique 1-dimensionnelle en lignes. Pour permettre aux outils de rendu/composition d'effectuer cette coupure de manière adéquate, on peut d'introduire des indices de sécabilité par le biais d'attributs de textème *ad hoc*. On distingue deux cas :

1. l'indice de sécabilité est constant ;

2. l'indice de sécabilité dépend du contexte, et l'opération de coupure peut entraîner des modifications dans les textèmes environnants.

Le premier cas est essentiellement celui de la coupure entre les *mots* ou entre les *sinogrammes*. Dans le cas des sinogrammes, la règle est simple : on peut couper après un caractère sinographique si et seulement s'il n'est pas suivi d'un signe de ponctuation. On peut modéliser cela en introduisant une « pénalité de coupure à gauche » LPENALTY, par exemple pour le petit dialogue < **你爱我？是。** > « tu m'aimes ? oui. » :

$$\begin{bmatrix}\text{CHAR} & \text{<你>}\\ \text{GLYPH} & \text{|你|}\end{bmatrix}\begin{bmatrix}\text{CHAR} & \text{<爱>}\\ \text{GLYPH} & \text{|爱|}\end{bmatrix}\begin{bmatrix}\text{CHAR} & \text{<我>}\\ \text{GLYPH} & \text{|我|}\end{bmatrix}\begin{bmatrix}\text{CHAR} & \text{<？>}\\ \text{LPENALTY} & 10000\\ \text{GLYPH} & \text{|？|}\end{bmatrix}$$

$$\begin{bmatrix}\text{CHAR} & \text{<是>}\\ \text{GLYPH} & \text{|是|}\end{bmatrix}\begin{bmatrix}\text{CHAR} & \text{<。>}\\ \text{LPENALTY} & 10000\\ \text{GLYPH} & \text{|。|}\end{bmatrix}$$

La coupure entre les mots (pour les écritures qui possèdent le concept de mot) peut également être gérée par l'attribut LPENALTY. Par exemple, en français, il est déconseillé de couper la ligne entre une abréviation comme « p. » et le nombre qui suit (par exemple |p. 5|). Cela peut être modélisé par la séquence d'AVM suivante :

$$\begin{bmatrix}\text{PHON} & \text{/paʒ/}\\ \text{ORTH} & \left\{\begin{bmatrix}\text{CHAR} & \text{<p>}\\ \text{GLYPH} & \text{|p|}\end{bmatrix}\begin{bmatrix}\text{CHAR} & \text{<.>}\\ \text{TYPE} & \textit{abr}\\ \text{GLYPH} & \text{|.|}\end{bmatrix}\right\}\\ \text{SYNSEM} & \begin{bmatrix}\text{CAT} & \textit{nom}\\ \text{GEN} & \textit{fém}\\ \text{NUM} & \textit{sing}\\ \text{SEM} & \textit{page}\end{bmatrix}\end{bmatrix}\boxed{\text{B}}\begin{bmatrix}\text{PHON} & \text{/sɛ̃k/}\\ \text{ORTH} & \left\{\begin{bmatrix}\text{CHAR} & \text{<5>}\\ \text{LPENALTY} & 10000\\ \text{GLYPH} & \text{|5|}\end{bmatrix}\right\}\\ \text{SYNSEM} & \begin{bmatrix}\text{CAT} & \textit{adjnum}\\ \text{POS} & \textit{post}\\ \text{SEM} & \textit{cinq}\end{bmatrix}\end{bmatrix}$$

où l'on a symbolisé par B l'espace intermots (la méthode traditionnelle pour indiquer l'insécabilité dans cette situation est d'utiliser un caractère spécial, l'« espace insécable », or cela voudrait dire que l'on traite des espaces intermots comme des graphèmes, ce qui est discutales, même s'il existe des caractères Unicode pour toute sorte d'espaces typographiques.

Degré de sécabilité et séquences graphétiques alternatives

Le deuxième cas de coupure est celui opéré à l'intérieur des mots, autrement dit : la *césure*. Il s'agit d'une opération qui mobilise des ressources linguistiques à plusieurs niveaux d'analyse [YH, 2006b, 2019f] :

- *graphémique* : dans plusieurs langues on forme des « syllabes écrites » et on coupe à leurs frontières. Ainsi, par exemple, en italien, le mot <attività> peut être coupé comme suit : <at·ti·vi·tà> ;
- *phonétique* : en grec moderne la séquence de graphèmes <ντ> est coupée lorsque sa réalisation phonétique est /nt/ ou /nd/, et insécable lorsqu'elle est /d/ : <ἐν·αν·τίον> /enand:ion/ mais <βί·ντε·ο> /v:ideo/ ;

- *morphologique* : en allemand la césure se fait à la frontière des composants (ainsi qu'à l'intérieur du dernier composant), par exemple : <Spät·an·ti·ke> (composants <Spät> et Antike) ;
- *syntaxique* : en anglais le mot <record> est coupé <rec·ord> lorsqu'il s'agit d'un nom et <re·cord> lorsqu'il s'agit d'un verbe ;
- *sémantique* : des césures comme *<the·rapist> [Knuth, 1984, p. 449] en anglais, *<con·science> en français ou *<Spargel·der> [Duden, 2018] en allemand obéissent à des règles de césure graphémiques et morphologiques mais posent un problème sémantique : de par le léger décalage temporel dû au passage à la ligne qui entraîne un accès intempestif au lexique metal [Nas, 1988] elles induisent le lecteur en erreur[27] et présentent un risque de perte du fil de la lecture.

Modéliser l'opération de césure consiste à injecter dans les textèmes deux types d'information :

1. le *degré de sécabilité*, autrement dit la priorité donnée aux différentes césures possibles ;
2. la *transformation de la séquence graphétique* dans l'éventualité d'une césure.

Le degré de sécabilité est binaire dans la plupart des langues (du moins pour les mots courts, puisqu'une césure, par exemple, du mot <anticonstitutionnel> après les deux premières lettres est autorisée, mais non conseillée), mais dans certaines langues, comme l'allemand, il existe une hiérarchisation des césures possibles (on coupe d'abord entre composants, ensuite à l'intérieur du dernier composant et enfin, en cas d'extrême nécessité, à l'intérieur des autres composants). Voici un exemple, où 0 signifie « pas de césure autorisée », 1 signifie « césure recommandée », 0,5 « césure autorisée mais non conseillée », 0,1 « césure autorisée uniquement en cas d'extrême nécessité » :

W a h r s c h e i n l i c h k e i t s t h e o r i e
0 0 0 1 0 0 0 0 0 0,1 0 0 0 0,1 0 0 0 0 1 0 0 0,5 0,5 0 0

En ce qui concerne le deuxième point, c'est-à-dire la transformation que subit la chaîne graphétique lors de la césure, il s'agit le plus souvent d'insérer un textème sans caractère qui correspond au trait de césure. Mais la transformation peut aller au-delà du simple ajout de trait de césure. Trois phénomènes peuvent se produire (par ordre décroissant de fréquence du phénomène) :

2.a un ajout de trait de césure ;
2.b la disparition d'une ligature et, le cas échéant, l'apparition d'une ou plusieurs nouvelles ligatures ;
2.c un changement qui va au-delà de l'allographie et qui implique le remplacement de certains graphes par des graphes d'autres graphèmes, ou l'ajout de nouveaux graphes qui proviennent de la version sous-jacente des morphèmes.

Dans les trois cas on utilisera la méthode suivante : on ajoutera un attribut HYPH avec deux sous-attributs : l'indice IND du graphème après lequel la césure aura lieu, et une chaîne de glyphes ALT correspondant à la transformation subie.

Le premier cas est le plus simple et le plus universel. En guise d'exemple, voici l'AVM du mot <couper> avec informations de césure :

[27] Dans le cas de <the·rapist> « thérapeute » le lecteur risque de lire <the rapist> « le violeur » ; dans le cas de <con·science> la première syllabe prise individuellement constitue un mot du registre vulgaire ; dans le cas de <Spargel·der>, alors que la décomposition en composants correcte est <Spar> « économie » suivi de *Gelder* (« argent » au pluriel), cette césure donne l'impression que le premier composant est <Spargel> « asperge » et entraîne donc une lecture erronée du mot.

$$\begin{bmatrix}
\text{PHON} & \text{/kupe/} \\
\text{ORTH} & \left\{ \begin{bmatrix} \text{CHAR} & \text{<c>}_1 \\ \text{GLYPH} & |\text{c}|_1 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<o>}_2 \\ \text{GLYPH} & |\text{o}|_2 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<u>}_3 \\ \text{GLYPH} & |\text{u}|_3 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<p>}_4 \\ \text{GLYPH} & |\text{p}|_4 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<e>}_5 \\ \text{GLYPH} & |\text{e}|_5 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<r>}_6 \\ \text{GLYPH} & |\text{r}|_6 \end{bmatrix} \right\} \\
\text{HYPH} & \left\{ \begin{bmatrix} \text{IND} & 3 \\ \text{ALT} & \left\{ |\text{c}|_1, |\text{o}|_2, |\text{u}|_3, |\text{-}|, |\text{p}|_4, |\text{e}|_5, |\text{r}|_6 \right\} \end{bmatrix} \right\} \\
\text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{ver tr} \\ \text{MODE} & \textit{inf} \\ \text{SEM} & \textit{couper} \end{bmatrix}
\end{bmatrix}$$

Pour illustrer le deuxième cas, nous prenons l'exemple du mot <affine>. Il possède une ligature |ffi|, mais lors d'une césure après |af|, une nouvelle ligature |fi| apparaît en début du mot |fine| :

$$\begin{bmatrix}
\text{PHON} & \text{/afin/} \\
\text{ORTH} & \left\{ \begin{bmatrix} \text{CHAR} & \text{<a>}_1 \\ \text{GLYPH} & |\text{a}|_1 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<f>}_2 \\ \text{GLYPH} & |\text{ffi}|_2 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<f>}_3 \\ \text{GLYPH} & ||_3 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<i>}_4 \\ \text{GLYPH} & ||_4 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<n>}_5 \\ \text{GLYPH} & |\text{n}|_5 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & \text{<e>}_6 \\ \text{GLYPH} & |\text{e}|_6 \end{bmatrix} \right\} \\
\text{HYPH} & \left\{ \begin{bmatrix} \text{IND} & 2 \\ \text{ALT} & \left\{ |\text{a}|_1, |\text{f}|_2, |\text{-}|, |\text{fi}|_3, |\text{n}|_5, |\text{e}|_6 \right\} \end{bmatrix}, \begin{bmatrix} \text{IND} & 4 \\ \text{ALT} & \left\{ |\text{a}|_1, |\text{ffi}|_2, |\text{-}|, |\text{n}|_5, |\text{e}|_6 \right\} \end{bmatrix} \right\} \\
\text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{adj} \\ \text{NUMB} & \textit{sing} \\ \text{SEM} & \textit{affine} \end{bmatrix}
\end{bmatrix}$$

Enfin, le troisième cas concerne l'allemand et le grec. En allemand, et plus particulièrement celui d'avant la reforme orthographique de 1996, lorsque le digraphe |ck| doit être coupé, il devient |k-k| (la raison étant phonétique : |ck| se prononce comme une consonne /k/ géminée). De même lorsque la forme sous-jacente d'un mot composé contient trois consonnes identiques (les deux provenant d'un composant, et le troisième étant la consonne initiale du composant suivant) alors l'orthographe d'avant la reforme veut que seules deux consonnes soient écrites[28], mais en cas de césure la troisième consonne réapparaît : |Schiffahrt| devient |Schiff-fahrt|, |Bettuch| devient |Bett-tuch| (lorsqu'il signifie « drap », mais |Bet-tuch| lorsqu'il signifie « châle de prière », puisque étymologiquement il n'y a que deux |t|). Enfin, en grec, le tréma indique que deux voyelles sont prononcées individuellement et ne forment pas de digraphe : |πλαϊνός| /plain'os/ mais |παιδί| /peð'i/ ; en cas de césure entre les deux voyelles du digraphe le tréma disparaît, puisqu'un digraphe ne peut être coupé : <πλα-ινός>.

Voici l'AVM du mot |backen|, avec césure :

[28] Christa Dürscheid nous signale qu'il y a des exceptions : le mot Betttruhe (Bett-truhe, meuble de lit, à distinguer de Bett-ruhe, repos au lit) était déjà écrit avec trois <t> même avant la reforme.

$$\begin{bmatrix} \text{PHON} & /\text{b'akken}/ \\ \text{ORTH} & \left\{ \begin{bmatrix} \text{CHAR} & <\text{b}>_1 \\ \text{GLYPH} & |\text{b}|_1 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & <\text{a}>_2 \\ \text{GLYPH} & |\text{a}|_2 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & <\text{c}>_3 \\ \text{GLYPH} & |\text{c}|_3 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & <\text{k}>_4 \\ \text{GLYPH} & |\text{k}|_4 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & <\text{e}>_5 \\ \text{GLYPH} & |\text{e}|_5 \end{bmatrix}, \begin{bmatrix} \text{CHAR} & <\text{n}>_6 \\ \text{GLYPH} & |\text{n}|_6 \end{bmatrix} \right\} \\ \text{HYPH} & \left\{ \begin{bmatrix} \text{IND} & 3 \\ \text{ALT} & \left\{ |\text{b}|_1, |\text{a}|_2, |\text{k}|_3, |\text{-}|, |\text{k}|_4, |\text{e}|_5, |\text{n}|_6 \right\} \end{bmatrix} \right\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{ver tr} \\ \text{MODE} & \textit{inf} \\ \text{SEM} & \textit{cuire} \end{bmatrix} \end{bmatrix}$$

Autres possibilités

On peut imaginer d'autres possibilités d'utilisation des textèmes, en voici une provenant de [YH et Bella, 2005a] : dans les écritures bicamérales, les lettres possèdent deux casses, elles peuvent être bas-de-casse ou capitales. On considère que les deux casses d'une même lettre correspondent à des graphèmes différents[29], d'ailleurs Unicode les code en tant que caractères différents (avec des règles de passage d'une casse à l'autre). Inclure l'information bas-de-casse/capitale dans un textème serait donc redondant.

Mais en réalité il y a bien plus que deux cas :

1. la « casse de titre » : les graphes utilisés lorsqu'un mot tout entier est écrit en capitales. En français cette casse est identique aux capitales, mais en grec une lettre capitale va garder son accent et son esprit potentiel alors qu'une lettre en casse de titre sera écrite sans accent ni esprit : |Ἄλλος| mais ||A|ΛΛΟΣ| (le tréma et le iota souscrit sont les seuls diacritiques qui sont maintenus en casse de titre, comparer |Ἄυλος| et ||AΥ|ΛΟΣ| [YH et Plaice, 1999c]) ;
2. la « capitale obligatoire » : en français le premier mot d'une phrase est capitalisé, comme dans <Il m'en a parlé.>, mais cette capitale est accidentelle : <Je répète : il m'en a parlé.>. Or dans certains cas la capitalisation permet la désambiguïsation et contribue donc à la pragmatique du mot : en écrivant, par exemple, le pronom « Il » systématiquement avec une majuscule dans un contexte de texte religieux, on se réfère à Dieu ;
3. le « bas-de-casse obligatoire » : les abréviations sont sensibles à la casse, et en allemand certaines abréviations sont mixtes (bas-de-casse/capitales) pour distinguer les initiales de substantifs (capitalisés) des initiales des autres types de discours. Ainsi dans <GmbH> les <m> et <b> indiquent une préposition (*mit* « avec ») et un adjectif (*beschränkter* « limité »), ils doivent donc rester en bas-de-casse quelque soit le contexte ;
4. la « casse inversée » : parmi les méthodes d'écriture inclusive en langue allemande il existe le *binnen-I* [YH et Dichy, 2019] qui consiste à inverser la casse de la lettre <i> du suffixe féminin pluriel des substantifs multi-genres. Par exemple, <StudentInnen> est l'équivalent de notre <étudiant·e·s>. Or quand ce même mot apparaît dans un titre, on inverse la casse et on écrit <STUDENTiNNEN>. On peut donc qualifier la casse de cette lettre comme « casse inversée » de son contexte.

[29] Comparer <paris>, le pluriel de *pari*, et <Paris>, la ville. Les graphes <p> et <P> commutent, il s'agit donc par définition de deux graphèmes différents.

Inclure ces informations dans un textème permet de lever des ambiguïtés et de traiter correctement les données textuelles en cas de changement de casse appliqué algorithmiquement.

1.6 Le cas de l'écriture chinoise

L'écriture chinoise est utilisée pour le chinois, le japonais, le coréen (de plus en plus rarement) et le viêtnamien (historiquement). Elle forme un système complexe. Plusieurs tentatives ont eu lieu pour la modéliser : Fujimura et Kagaya [1969] et Wang [1983] ont utilisé la grammaire générative, Dürst [1993] la programmation logique avec une syntaxe de type Prolog, Chu [2003] la métaphore biologique des gènes et de l'ADN, Moro [2003] une approche orientée objets, Sproat [2000] le formalisme AVM avec cinq opérateurs de concaténation, Peebles [2007] et Bishop et Cook [2007] des arborescences XML symboliques et géométriques, et Qin et al. [2002] la combinatoire des douze opérateurs Unicode IDS (*Ideographic Description Characters*). Un certain nombre de travaux ont envisagé l'écriture chinoise en tant que réseau : Fujiwara et al. [2004] ; Li et Zhou [2007] (les arêtes étant des composants qui coexistent dans le même sinogramme), Rocha et Fujisawa [1996] (graphe biparti de composants et des sinogrammes), Zhou et Liu [2002] (où les arêtes sont des sinogrammes qui coexistent dans des mots), Yu et al. [2011] (réseau phonémique), Li et Zhou [2007] (graphe biparti des composantes phonétiques et sémantiques), et Hsieh [2006], Chou et al. [2007] (reliant les inclusions de composants à des relations dans des ontologies). La communauté de psychologie cognitive s'est également intéressée au sujet Taft et Zhu [1997] ; Williams et Bever [2010], et en particulier à l'incidence de l'ordre des traits et des radicaux sur la compétence linguistique [Tamaoka et Yamada, 2000].

Notre approche [YH, 2011, 2013], qui a été citée dans [Myers, 2019] (qui est actuellement le summum et la synthèse des travaux sur l'écriture chinoise) a consisté à quantifier la phonéticité et la sémanticité au niveau du graphe des inclusions de composants. Nous la décrivons dans la section suivante.

1.6.1 Modélisation

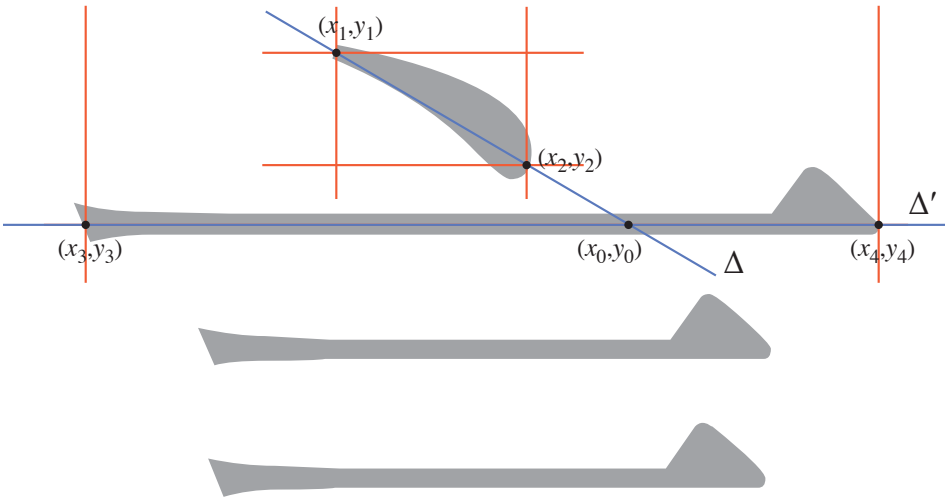
Les sinogrammes sont des combinaisons graphiques de trente-six traits fondamentaux contenues dans un carré parfait (appele *carré sinographique*). Les traits sont tracés dans un ordre immuable et connu de tout scripteur de l'écriture chinoise, l'*ordre traditionnel des traits*. Dans l'histoire de cette écriture il y a eu un certain nombre de tentatives de regroupement des traits pour former des composants. Ces composants, appelés *radicaux* ou *clés*, ont été utilisés dès le 1^{er} siècle av. J.-C. pour classer les sinogrammes dans les dictionnaires.

De notre côté, nous avons choisi de faire table rase des systèmes de clés et de décrire les sinogrammes graphiquement, par le biais de paires consécutives de traits. Comme la taille relative des traits peut varier (par exemple, comparons 土 « sol » et 士 « soldat » qui comportent tous les deux des paires de traits horizontaux, mais avec des ratios de largeur différents) nous avons introduit le système de modélisation suivant :

1. chaque trait fondamental est noté par une combinaison de lettres (`d` pour « point », `h` pour « trait horizontal », etc.) ;
2. chaque trait fondamental possède un axe Δ, c'est-à-dire une droite affine qui reste relativement stable lorsque le trait se trouve dans des allographes stylistiques ;

3. les points d'intersection entre Δ et l'enveloppe rectangulaire du trait sont appelés α et β (ils sont pris dans l'ordre de tracé du trait) ;
4. la modélisation d'un sinogramme à n traits consiste en une suite $(t_1, N_1, t_2, N_2, t_3, \ldots, t_{n-1}, N_{n-1}, t_n)$ où les t_i sont les symboles des traits (dans l'ordre traditionnel des traits) et N_i des quadruplets de nombres rationnels.

Voici la sémantique de ces nombres, illustrée par un exemple :
La première paire de traits de la figure



(où les lignes bleues Δ et Δ' représentent les axes des traits et les lignes rouges les enveloppes rectangulaires) est du type $t_1 = \texttt{d}$, $t_2 = \texttt{h}$ avec $N_1 = (1.4, 9.1, 0, 0.6)$ calculé comme suit :

1. la première valeur est $\frac{d((x_0,y_0)-(x_1,y_1))}{d((x_2,y_2)-(x_1,y_1))}$, autrement dit la distance entre le point α du premier trait et l'intersection des axes des deux traits, en prenant la distance $d(\alpha,\beta)$ du premier trait comme unité de longueur,
2. la deuxième valeur est $\frac{|x_4-x_3|}{|x_2-x_1|}$, c'est-à-dire le ratio des largeurs des enveloppes rectangulaires des deux traits,
3. la troisième valeur est $\frac{|y_4-y_3|}{|y_2-y_1|}$, c'est-à-dire le ratio des hauteurs des enveloppes rectangulaires des deux traits,
4. la quatrième valeur est $\frac{d((x_0,y_0)-(x_3,y_3))}{d((x_4,y_4)-(x_3,y_3))}$, autrement dit la distance entre son point α du deuxième trait et l'intersection des axes des deux traits, en prenant la distance $d(\alpha,\beta)$ du deuxième trait comme unité de longueur.

À noter que certaines parmi ces valeurs peuvent être infinies, on écrira alors le symbole E.

Une fouille des descriptions $t_1, N_1, \ldots, t_{n-1}, N_{n-1}, t_n$ des sinogrammes du corpus Wenlin de la totalité des sinogrammes Unicode encodés en CDL Bishop nous a permis de dégager des motifs fréquents qui jouent le rôle de composants. Nous nous sommes intéressés à un type particulier de composants, ceux qui existent aussi en tant que sinogrammes individuels et de ce fait ont (au moins) une sémantique et une représentation phonétique. On les appelle des *sous-caractères*.

Un autre problème non négligeable posé par l'écriture chinoise est celui de l'allographie due à la disparité géographique (nous avons vu à la p. 15 les six formes de base associées au sinogramme 伶)

mais aussi, et surtout, aux réformes, la réforme la plus connue étant celle qui a eu lieu en Chine continentale en 1958 (avec une deuxième étape en 1986) et qui a simplifié quelques 1 753 sinogrammes. Étant donné qu'un sinogramme simplifié a exactement la même sémantique et la même phonétique que le sinogramme original, nous ne pouvons que les considérer comme des allographes du même graphème, même si leurs graphes peuvent être sensiblement différents (comme, par exemple, dans le cas du 葡 qui est devenu 𠬜). Nous avons donc défini des *classes allographiques* qui comportent toutes les formes de base associées à un sinogramme lorsqu'on tient compte des disparités géographiques et des conséquences des réformes. Pour ne traiter que les sinogrammes pour lesquels nous disposions d'informations phonétiques et sémantiques, nous nous sommes limités à l'étude de 18 686 classes allographiques, dont 87% étaient des singletons (un seul graphe) et la moyenne du nombre d'allographes est de 1,14 par classe.

Nous définissons une *inclusion* (notée $s \to c$) comme une paire (s, c) où s et c sont des sinogrammes et s est sous-caractère de c. En relevant les inclusions au niveau des classes allographiques, nous obtenons des *inclusions de classes*. Sur la fig. 1.8, p. 40, nous présentons une représentation graphique des inclusions : un graphe dont les sommets sont les 18 686 classes allographiques et dont les arêtes sont les 39 719 inclusions de classes observées.

1.6.2 Phonéticité et sémanticité

En linguistique, et plus particulièrement en sinographématique [Schindelin, 2007], la phonéticité et la sémanticité d'un composant sont définies comme la proximité phonétique (resp. sémantique) du composant vis-à-vis du sinogramme dont il fait partie. Ainsi, par exemple, Sproat décrit l'AVM du sinogramme 鮫 comme suit :

$$\begin{bmatrix} \text{PHON} & \begin{bmatrix} \text{SYL} & \begin{bmatrix} \text{SEG} & \text{<[ONS /ji/][RIME /ao/>} \\ \text{TONE} & 1 \end{bmatrix} \end{bmatrix}_{2^*} \\ \text{ORTH} & \{\text{魚}_1, \text{交}_2\} \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{nom} \\ \text{SEM} & \textit{requin}_{1^*} \end{bmatrix} \end{bmatrix}$$

en décomposant 鮫 en 魚 et 交, et en indiquant que le premier est le « composant sémantique » 鮫 signifie « requin » alors que 魚 signifie « poisson », un cas classique d'hypéronymie) et le deuxième est le « composant phonétique » (aussi bien 交 que 鮫 se prononcent /jiāo/ en mandarin).

Nous avons poussé ces notions plus loin en définissant des *coefficients phonétique et sémantique*.

Pour définir le *coefficient phonétique* nous nous sommes munis de distances phonétiques d pour deux langues sinographiques (Chang et al. [2010] pour le mandarin, notre propre définition pour le japonais) et nous avons défini le coefficient phonétique d'une sous-classe allographique $\mathbf{s}$ vis-à-vis de la classe $\mathbf{c}$ du sinogramme comme étant

$$\phi(\mathbf{s}, \mathbf{c}) := 1 - \frac{\min_{s \in \mathbf{s}, c \in \mathbf{c}} d(s, c)}{\max_{\mathbf{s},\mathbf{c} \text{ tels que } s \to c} \min_{s \in \mathbf{s}, c \in \mathbf{c}} d(s, c)}.$$

Le dénominateur (qui ne dépend que de la langue choisie) entraîne le fait que $\text{Im}\phi \subset [0, 1]$.

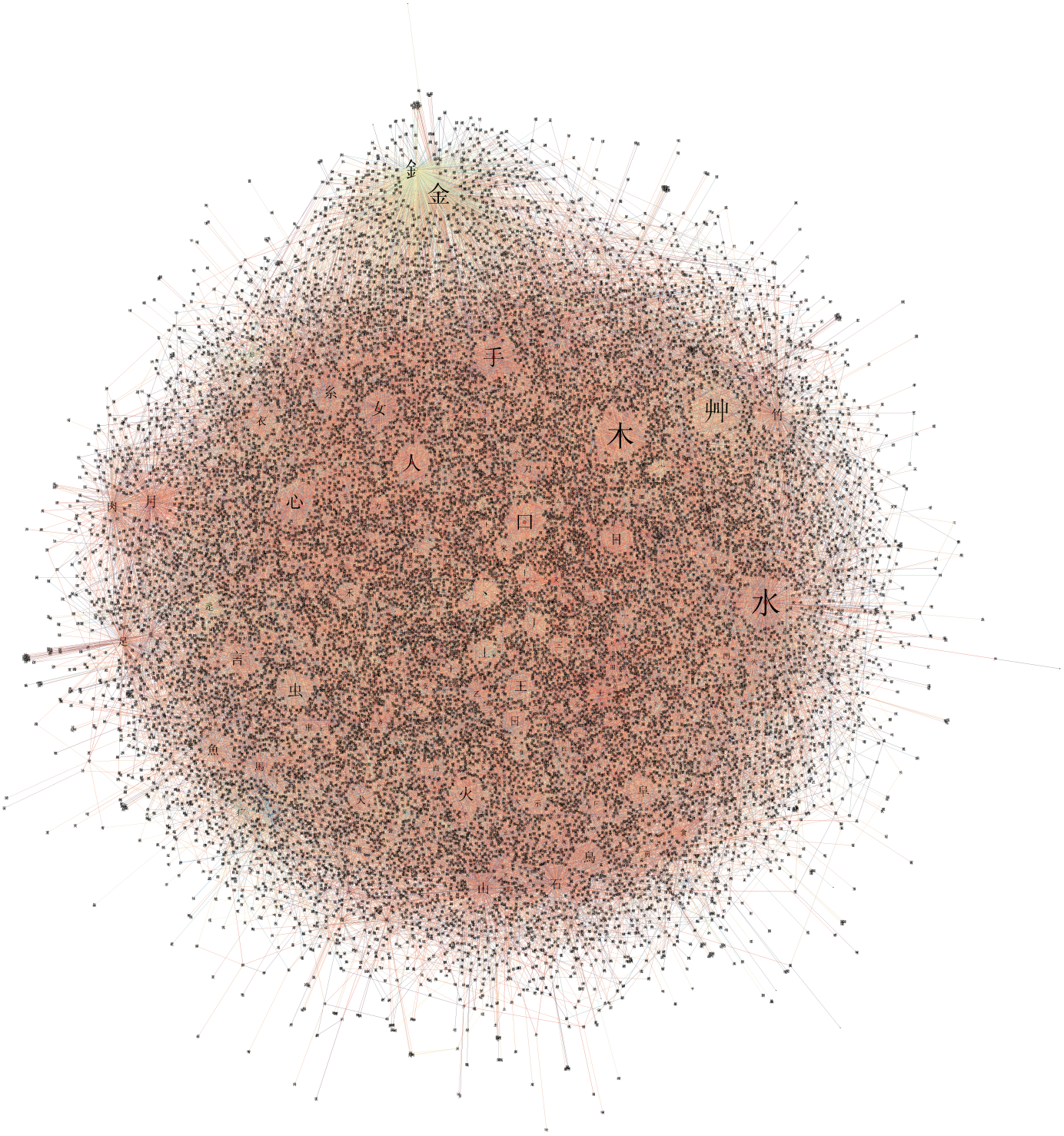


FIG. 1.8: Représentation visuelle du graphe des inclusions.

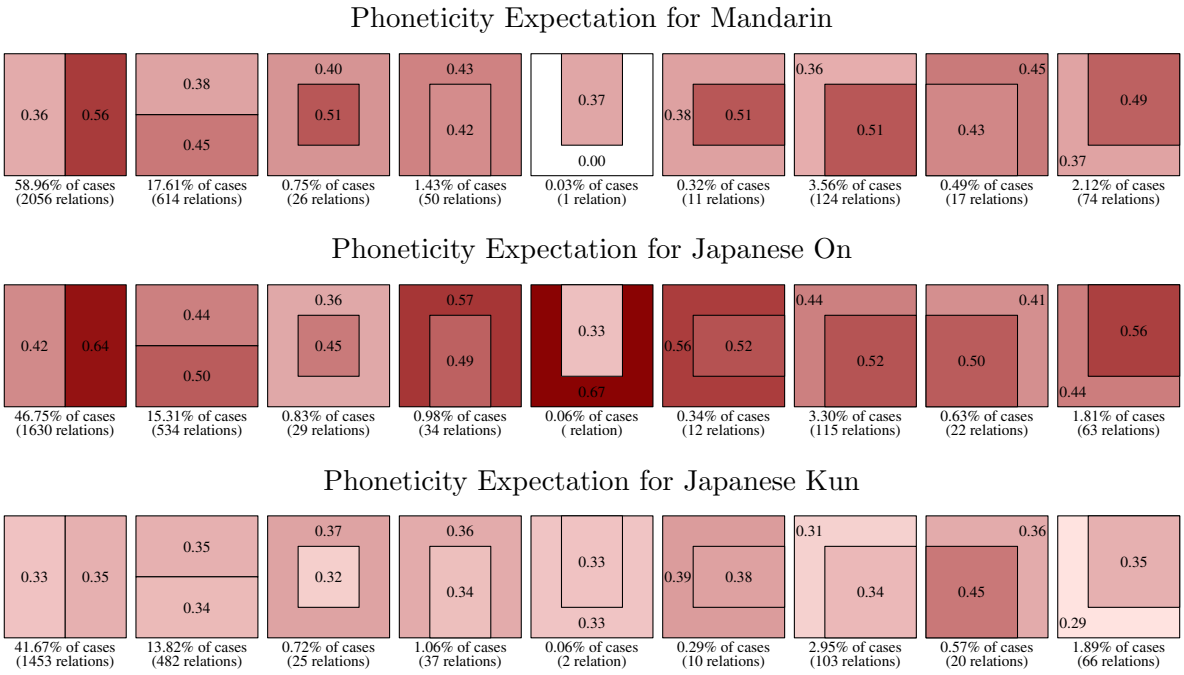


FIG. 1.9: Espérance de phonéticité par langue pour chaque type de sinogramme binaire.

Pour les sinogrammes binaires, c'est-à-dire composés d'exactement deux composants, nous avons calculé l'*espérance de phonéticité*, définie comme suit :

$$E_\phi(S) = \sum_{\mathbf{s},\mathbf{c} \text{ tels que } \mathbf{s}\to\mathbf{c}} \phi(\mathbf{s},\mathbf{c}) P(\phi(\mathbf{s},\mathbf{c}) = x).$$

Les résultats sont présentés sur la fig. 1.9. *On* et *kun* sont les deux classes de réalisation phonétique des sinogrammes japonais : la première est celle importée de Chine en même temps que les sinogrammes, il s'agit donc de mots chinois dont la prononciation a été simplifiée pour s'adapter à la phonologie du japonais ; la deuxième est celle des mots d'origine japonaise. Il est remarquable que pour le premier type de sinogrammes (deux composants en forme de demi-carrés verticaux alignés horizontalement), un type qui représente 59% des sinogrammes examinés en mandarin et 47% des sinogrammes examinés en japonais *on*, la phonéticité accrue du composant droit est manifeste. Le fait qu'elle est plus manifeste en japonais qu'en mandarin vient peut-être du fait que, ayant été adaptées au système phonologique du japonais, les réalisations phonétiques du sous-caractère et du sinogramme finissent par être plus proches (en particulier, il n'y a pas de ton en japonais). Voici un exemple qui illustre ce propos :

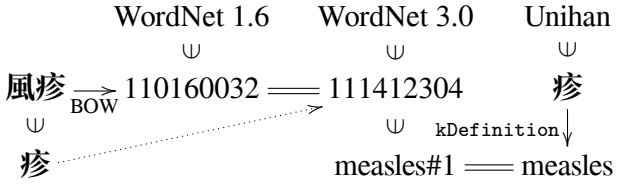
	Mandarin	Jap. On	Jap. Kun
任	rèn	nin	makaseru, ninau, taeru
↑	↑≈	↑=	↑≠
人	rén	nin	hito

On voit sur ce diagramme que pour les sinogrammes 任 « responsabilité » et 人 « homme », en mandarin il y a juste une différence de ton alors qu'en japonais *on* les deux ont exactement la même réalisation phonétique. En japonais *kun* les deux réalisations phonétiques sont totalement différentes.

Pour définir le coefficient sémantique nous nous sommes basés sur des ressources linguistiques : la ressource *Academia Sinica BOW* [Huang, 2003] pour le chinois en écriture traditionnelle, le WordNet chinois [Gao et al., 2008] pour le chinois en écriture simplifiée et le WordNet japonais [Isahara et al., 2008]. Toutes trois possèdent des liens vers les identifiants de synset du WordNet anglais (v1.6 pour la première, v2 pour la deuxième, v3 pour la troisième).

Les entrées à un seul sinogramme dans ces ressources s'élèvent à 3 075 pour le chinois traditionnel, 2 440 pour le chinois simplifié et 4 941 pour le japonais. La base de données Unihan[30], développée et maintenue par le Consortium Unicode, possède un grand nombre de sémantiques affectées aux sinogrammes, mais sans l'aspect formel des identifiants WordNet. Nous avons utilisé la méthode suivante pour combiner la richesse d'Unihan et la précision des ressources linguistiques du paragraphe précédent : soit w_i un mot chinois ou japonais possédant l'ID σ_i dans l'un des trois WordNet ; soit $e_{i,k}$ l'un des termes du synset du WordNet anglais avec le même ID σ_i ; si $e_{i,k}$ existe dans Unihan en tant que sémantique d'un sinogramme (isolé) c_j et si $c_j \in w_i$ alors on attache l'ID σ_i à c_j. Nous relevons cette information au niveau des classes allographiques.

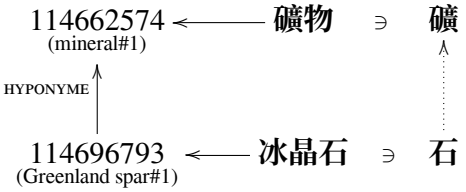
En voici un exemple :



Puisque **疹** a le sens « measles » (« rubéole ») dans Unihan et est contenu dans le mot **風疹** qui appartient au synset de la rubéole (« measles#1 »), nous attachons l'identifiant synset de la rubéole à **疹** .

Grâce à cette méthode nous avons pu annoter sémantiquement (par un identifiant de synset WordNet) 1 392 classes allographiques supplémentaires, arrivant ainsi à un total de 4 244 classes annotées.

Pour obtenir des relations sémantiques nous avons appliqué la méthode suivante : soit w un mot appartenant à un synset σ et w' un mot appartenant à un synset σ'. Imaginons que σ est un hyponyme de σ'. Soit c un caractère de w et c' un caractère de w' et imaginons que w est un sous-caractère de w'. Alors on décrète qu'il existe une relation sémantique entre c et c'. Exemple :



Nous avons comptabilisé le nombre de telles relations sémantiques entre deux sinogrammes et obtenu ainsi un poids de relation sémantique. Au total nous avons pu répertorier 6 816 inclusions donnant lieu à une relation sémantique.

1.6.3 Applications à la fouille de texte et évaluation

Le but de la modélisation de l'écriture chinoise a été de récupérer de l'information à partir des composants des sinogrammes pour améliorer les traitements automatiques de la langue. En effet,

[30] `https://unicode.org/charts/unihan.html`

pour les algorithmes de TAL l'unité atomique du chinois est le sinogramme et aucune attention n'est apportée aux composants. Nous nous sommes posés la question si l'ajout d'information concernant les composants pouvait améliorer les traitements automatiques et pour y répondre nous avons effectué une tâche de classification de textes.

Dans la suite nous allons décrire les corpus utilisés pour l'évaluation, ainsi que la manière dont l'information sur les composants a été attachée aux sinogrammes.

Nous avons utilisé un corpus de 20 000 articles en ligne chinois (corpus *Sogou of Chinese News*) d'un total de 14 millions de caractères Unicode, comportant quatre classes de même taille (sports, finance, actualités, loisirs) ainsi qu'un corpus de même taille de brèves de l'agence Reuters japonaise, comportant les mêmes quatre classes, d'un volume global de 4 millions de caractères Unicode. Nous avons enlevé tout caractère non-sinographique (y compris les syllabes kana dans le corpus japonais) ainsi que tout sinogramme d'une fréquence inférieure à 10. Après ce filtrage restaient 4 275 sinogrammes distincts dans le corpus chinois et 2 010 dans le corpus japonais. Nous avons appliqué un SVM linéaire avec validation 10-croisée et avons obtenu les résultats de justesse (*accuracy*) suivants :

	Justesse	Nb de vecteurs de support
Chinois	89,6605%	4 933
Japonais	86,925%	4 237

Ces valeurs nous ont servi de résultats de base, sans utilisation de sous-caractères.

Nous avons vu ci-dessus comment calculer les coefficients de phonéticité et de sémanticité $\phi(s,c)$ (resp. $\sigma(\mathbf{s},\mathbf{c})$) d'une inclusion de classes. Nous définissons :

1. la *chaîne la plus sémantique* comme la suite $\mathbf{s}_1 \to \cdots \to \mathbf{s}_n \to \mathbf{c}$ de classes allographiques de sous-caractères telles que chaque inclusion $\mathbf{s}_{i-1} \to \mathbf{s}_i$ soit de sémanticité maximale parmi toutes les classes de sous-caractères de $\mathbf{s}_i$;
2. la *chaîne la moins phonétique* comme la suite $\mathbf{s}_1 \to \cdots \to \mathbf{s}_n \to \mathbf{c}$ de classes allographiques de sous-caractères tels que chaque inclusion $\mathbf{s}_{i-1} \to \mathbf{s}_i$ soit de phonéticité minimale parmi toutes les classes les sous-caractères de $\mathbf{s}_i$.

Pour toutes les instances d'une classe $\mathbf{s}_i$, nous choisissons un représentant dont la fréquence normalisée dans le corpus est maximale. On appelle cette fréquence le *poids* du sinogramme.

On utilise deux stratégies : dans la première on ajoute les représentants des classes $\mathbf{s}_i$ de la chaîne la plus sémantique avec un poids égal à $\frac{1}{n-i+1}\sigma(s_{i-1} \to s)$ (le poids devenant de plus en plus faible lorsqu'on s'éloigne de c dans la chaîne). Dans la deuxième stratégie on ajoute à la première également les représentants des classes de la chaîne la moins phonétique, avec un poids égal à $\frac{1}{n-i+1}\phi(s_{i-1} \to s)$. Voici les résultats obtenus :

Stratégie	Corpus	Justesse	# de vect. de suport
1	Chinois	**92,62**%	3 287
1	Japonais	89,99%	3 728
2	Chinois	92,435%	3 299
2	Japonais	**90,125**%	3 737

On constate une amélioration de la justesse par rapport aux résultats de base : de 89,6% on est passé à 92,62% pour le chinois et de 86,925% à 90,125% pour le japonais, ce qui signifie une réduction de 28,8% des erreurs de classification pour le chinois et de 24,5% pour le japonais.

1.7 Perspectives

Un article sur les *Méthodes graphétiques et graphématiques dans la fiction spéculative* est en cours d'écriture, il sera publié dans les *Actes du II^e colloque « Grapholinguistics in the 21st Century », 2020*. Dans cet article nous étendons l'approche graphétique de Meletis [2015, 2019] aux diverses parties du modèle traditionnel du livre et nous faisons un inventaire des lieux où une innovation/transgression graphé(ma)tique est envisageable. Lorsque de tels cas ont été constatés, nous les illustrons par des exemples tirés de la littérature de fiction spéculative. Une attention particulière est apportée aux traductions et à la manière dont les particularités graphétiques et graphématiques sont traduites d'une langue à l'autre.

Un projet en cours est l'étude de la *dynamique des tracés de sinogrammes* pour mesurer des infimes pauses entre les tracés de composants (dans le cadre d'un même sinogramme) ou les tracés de morphèmes (dans le cadre d'une séquence graphétique 1-dimensionnelle). Une société rennaise partenaire a développé une application pour tablette qui enregistre la dynamique des tracés (courbes horodatées au millième de seconde près) et permet aux scripteurs du japonais ou du chinois d'écrire des séquences de sinogrammes dont le contenu est présélectionné. Nos collègues James Myers à Taïwan, Terry Joyce à Tokyo et Hisashi Masuda à Hiroshima vont procéder à la collecte du corpus que nous allons ensuite analyser. Ce projet est inspiré des travaux de Ballier et al. [2019] et de Weingarten et al. [2004] qui ont mesuré les intervalles entre les frappes de touches de clavier et ont découvert que, pour l'allemand, ces intervalles sont plus importants entre les morphèmes, moins importants entre les syllabes, et encore moins importants entre les lettres. Comme la saisie du chinois et du japonais se fait par le biais de différentes méthodes de saisie [Lunde, 2008, p. 193–362] basées sur la transcription phonétique des sinogrammes mais aussi sur leur répétition dans le texte (les sinogrammes récemment rencontrés étant placés plus haut dans la liste des candidats), les méthodes d'horodatage de touches de clavier ne peuvent s'appliquer aux langues sinographiques et c'est pour cette raison que nous avons opté pour l'étude du tracé manuel.

Un autre projet en cours concerne la *profondeur orthographique* [van den Bosch et al., 1994 ; Borleffs et al., 2017] et une nouvelle méthode pour la mesurer. En effet, Zhang et al. [2015] décrivent un réseau neuronal convolutif qui est capable d'effectuer des tâches de classification de textes en utilisant comme corpus d'apprentissage des textes bruts sans aucune annotation linguistique. Cet article laisse penser que le réseau de neurones peut « retrouver tout seul toute la structure linguistique du texte » et arrive à effectuer des tâches avec des performances comparables aux méthodes traditionnelles sans avoir besoin de ressources linguistiques. Notre idée est de mesurer les performances du réseau sur des variantes orthographiques du même texte pour tester l'hypothèse que *plus une orthographe est profonde, meilleures seront les performances du réseau*. Ainsi, nous comparerons le japonais standard avec le japonais écrit en syllabique, l'arabe avec et sans voyelles, les grecs monotonique et polytonique, différentes propositions de simplification orthographique de l'anglais, etc.

Enfin, dans les années à venir, nous avons l'intention d'œuvrer pour la convergence des différents modèles de l'écriture : le modèle (grapho)linguistique [Meletis, 2019], les modèles informatiques basés sur Unicode, TEX ou XSL-FO [YH, 2004a] et le modèle typographique [Wehde, 2000], et ceci pour tous les systèmes d'écritures et toutes les traditions typographiques, y compris celles du Moyen-Orient et de l'Extrême-Orient.

Chapitre 2

Langages visuels et hybrides

Selon Marriott et Meyer [1998, p. 2], un *langage visuel* est

> un ensemble de diagrammes qui sont des « phrases » valides dans ce langage. Un tel diagramme est une collection de « symboles » dans un espace 2- ou 3-dimensionnel. La validité des phrases dépend de l'arrangement spatial des symboles. Le sens d'une phrase est, en général, constitué par les symboles graphiques utilisés dans la phrase et leur arrangement spatial.

Il y a différents formalismes permettant de définir et d'utiliser les langages visuels : des grammaires formelles spéciales (comme celle de [Sproat, 2000], vue au §1.3.4), la logique du 1er ordre, les graphes conceptuels, les logiques de description, etc.

Nous allons décrire deux applications des langages visuels : (a) la modélisation des figures géométriques euclidiennes à l'aide de graphes conceptuels étendus, dans le but de permettre la fouille de figures géométriques et la démonstration automatique de théorèmes, et (b) la preuve de concept d'un langage hybride (visuel/textuel) pour interagir avec une base de connaissances de navigation maritime du Service hydrographique et océanographique de la Marine.

2.1 Figures géométriques et graphes conceptuels

Une figure euclidienne, c'est-à-dire tracée « à la règle et au compas »[1], peut être considérée comme un *énoncé dans un langage formel écrit* dont les membres de l'alphabet sont des points, des lignes droites, des cercles, etc. Étant donné que la concaténation des ces éléments de base s'opère dans l'espace et que leurs formes peuvent varier, nous avons bien à faire à un langage visuel dont les instances d'éléments géométriques sont des allographes. Pour explorer ce langage visuel nous allons utiliser deux formalismes : celui de logique du 1er ordre [Chou et al., 2000] et celui de graphe conceptuel [YH et Quaresma, 2018b].

La formalisation en logique du premier ordre de la géométrie euclidienne a permis la preuve par ordinateur de plusieurs théorèmes ainsi que la découverte de nouveaux théorèmes. D'autre part il existe de plus en plus de figures géométriques en ligne Quaresma2011 (créées souvent par des logiciels éducatifs en ligne tels que *Geogebra*) mais aucun moteur de recherche permettant de les

[1] Euclide n'a jamais parlé dans ses *Éléments* de « figure tracée à la règle et au compas ». Néanmoins deux de ses cinq postulats correspondent aux usages traditionnels de ces deux instruments : le premier postulat (*Entre deux points on peut toujours tracer une droite*, Ἠιτήσθω ἀπὸ παντὸς σημείου ἐπὶ πᾶν σημεῖον εὐθεῖαν γραμμὴν ἀγαγεῖν) pour la règle et le troisième postulat (*Partant d'un point et d'une longueur donnés, on peut toujours tracer un cercle*, Καὶ παντὶ κέντρῳ καὶ διαστήματι κύκλον γράφεσθαι) pour le compas.

retrouver, la difficulté étant, entre autres, qu'une même figure peut résulter de plusieurs descriptions de départ et qu'une série d'inférences peut être nécessaire pour établir l'équivalence logique des descriptions (par exemple, un triangle équilatéral peut être défini de manière équivalente comme un triangle dont les trois côtés sont égaux ou comme un triangle dont les trois angles sont égaux, ou comme un triangle dont les trois hauteurs sont égales). Il existe donc un besoin de formalisation des figures géométriques permettant la fouille et capable d'intégrer suffisamment d'informations dans une même description de figure pour minimiser le recours aux inférences.

2.1.1 Univers géométrique de Chou-Gao-Zhang

Soit Σ une signature typée et E une Σ-interprétation. On appelle (Σ, E) un *univers géométrique* [Mathis et Thierry, 2010, §2.1]. Soit FOL$(\exists, \wedge, \Sigma)$ le fragment positif (sans négations), existentiel et conjonctif de la logique du 1$^{\text{er}}$ ordre typée et basée sur Σ. On appelle *système de contraintes* un triplet $S = (C, X, A)$ où C est un ensemble de formules bien formées de FOL$(\exists, \wedge, \Sigma)$ appelées *faits*, X est un ensemble de variables de Σ appelées *variables géométriques* et A est un ensemble de variables de Σ appelées *paramètres* tels que les arguments des prédicats de C sont soit dans $X \cup A$, soit des constantes de Σ.

Si $\varrho : A \to E$ est une application compatible avec les types, une *figure obtenue à partir du système de contraintes* S est une application compatible avec les types $f_\varrho : X \to \mathcal{E}$ telle que $\mathcal{E}$ soit un modèle de C.

Un *ensemble de règles géométriques* est un ensemble $\mathcal{R}$ de règles d'inférence sur Σ, de la forme $\bigwedge_i P_i \vdash \bigwedge_j Q_j$ ou $\bigwedge_i P_i \vdash \exists X \bigwedge_j Q_j$. Elles sont utilisées afin d'ajouter des nouveaux faits à S.

On appelle *théorie géométrique* générée par S la clôture inférentielle $\overline{S}$ de S obtenue en appliquant les règles de $\mathcal{R}$ jusqu'à aboutir à un point fixe, c'est-à-dire à un état de la théorie où l'application des règles d'inférence n'apporte aucun nouveau fait. La *figure géométrique obtenue par S* est la figure obtenue par le système de contraintes S.

Dans la suite nous allons présenter un premier univers géométrique, celui de Chou-Gao-Zhang (abréviation CGZ) [Chou et al., 2000]. La signature Σ_{CGZ} de cet univers utilise un seul type et 12 prédicats, elle est définie comme suit :

- types : point ;
- prédicats ([Chou et al., 2000, p. 228]) :

symbole	explication	arité	signature
points	existence de points	n	point $\times\, n$
coll	colinéarité	n	point $\times\, n$
para	parallélisme	$4n$	(point,point) $\times\, 2 \times n$
perp	perpendicularité	$4n$	(point,point) $\times\, 2 \times n$
midp	milieu de segment	3	point $\times\, 3$
cyclic	cocyclicité	n	point $\times\, n$
circle	cocyclicité avec centre donné	$n+1$	point, point $\times\, n$
eqangle	égalité d'angles	$8n$	(point,point) $\times\, 2 \times 2 \times n$
cong	congruence de segments	2n	(point,point) $\times\, n$
eqratio	égalité de ratios	$8n$	(point,point) $\times\, 2 \times 2 \times n$
simtri	similarité de triangles	$3n$	(point,point,point) $\times\, n$
contri	congruence de triangles	$3n$	(point,point,point) $\times\, n$

où les lignes sont représentées par des paires de points ; dans midp(M, A, B) le point M est le milieu

du segment AB ; dans $\mathsf{circle}(O, P_1, \ldots, P_n)$ le point O est le centre du cercle ; eqangle compare les angles pleins[2] formés par des paires de lignes $\ell_1, \ell_2, \ell_3, \ell_4$ (et donc par des quadruplets de paires de points) ; eqratio prend comme arguments des quadruplets de segments (et donc des octuplets de points) s_1, s_2, s_3, s_4 tels que $s_1/s_2 = s_3/s_4$; enfin, pour toutes ces définitions on prend soin d'appliquer des conditions de non-dégénérescence.

La théorie géométrique sur laquelle se base l'approche de CGZ est constituée de 93 règles d'inférence, dont 75 sont de la forme $P_1 \wedge \cdots \wedge P_n \vdash Q_1 \wedge \cdots \wedge Q_m$ où P_i et Q_j sont des prédicats de Σ_{CGZ} et 18 règles du type $P_1 \wedge \cdots \wedge P_n \vdash \exists X\, Q_1 \wedge \cdots \wedge Q_m$ où P_i et Q_j sont prédicats et X est une variable de Σ_{CGZ}. Les premières servent à décrire les relations entre points (et en particulier gèrent la symétrie de constructions telles que le parallélisme ou la cocyclicité) alors que les deuxièmes introduisent des nouveaux points (par exemple des intersections de droites, qui doivent exister si les conditions de non-dégénérescence sont respectées, etc.).

Nous donnons dans [YH et Quaresma, 2018a] les étapes de la démonstration dans l'univers CGZ du théorème de l'orthocentre (le fait que les trois hauteurs d'un triangle passent pas le même point), il nous a fallu pour cela 68 applications de règles géométriques, dont 72% étaient des règles de permutation d'arguments. Parmi ces applications de règles, seules trois représentent vraiment des faits géométriques importants : (D42) quand deux points sont vus par d'autres points avec les mêmes angles, alors les quatre points sont cocycliques et réciproquement (D41), (D21) quand deux angles pleins $\angle[\ell_1, \ell_2]$ and $\angle[\ell_3, \ell_4]$ sont égaux, alors les angles $\angle[\ell_1, \ell_3]$ and $\angle[\ell_2, \ell_4]$ sont égaux aussi.

2.1.2 Univers géométrique de Haralambous-Quaresma

Dans le but d'interpréter les figures euclidiennes de manière plus proche à la représentation des connaissances nous introduisons dans [YH et Quaresma, 2018b,a] un autre univers géométrique, avec des caractéristiques différentes de celui ce Chou-Gao-Zhang. En voici la signature :

- types : p (point), s (segment), l (ligne), c (cercle), a (angle), r (ratio) ;
- constantes : a_0 (angle plein $\angle[0]$), a_1 (angle plein $\angle[1]$), r_1 (ratio 1) ;
- prédicats :

symbole	explication	arité	signature
extr	le point est une extrémité du segment	2	p, s
cont	le segment est contenu dans la ligne	2	s, l
inci_l	le point est incident à la ligne	2	p, l
inci_c	le point est incident au cercle	2	p, c
center	le point est centre du cercle	2	p, c
angle	l'angle est défini par deux lignes	3	a, l, l
summit	le point est le sommet de l'angle	2	p, a
ratio_s	le ratio est défini par deux segments	3	r, s, s
ratio_a	le ratio est défini par deux angles	3	r, a, a

Nous appelons cette signature Σ_{HQ}. Nous décrivons dans [YH et Quaresma, 2018a] une application

[2] Nous appelons « angle plein » (*full-angle*) et nous notons par $\angle[u, v]$ une valeur numérique unique calculée pour toute paire de droites, dont la définition repose sur 14 contraintes données dans [Chou et al., 1996, p. 351] (par exemple le fait que si deux droites sont parallèles, leur angle plein est nul $\angle[0]$; que quand elles sont perpendiculaires, leur angle plein est égal à 1, noté $\angle[1]$; qu'il existe une opération nommée « addition » entre angles pleins qui est commutative et associative avec $\angle[0]$ comme élément neutre ; que $\angle[1] + \angle[1] = \angle[0]$; que la fonction $\angle[.,.]$ est antisymétrique ; et ainsi de suite). L'avantage de l'angle plein est que l'on n'a pas à se soucier d'« angle interne » (aigu) et d'« angle externe » (obtus) entre deux droites.

réversible $\Sigma_{\mathrm{CGZ}} \to \Sigma_{\mathrm{HQ}}$ qui envoie chaque prédicat de Σ_{CGZ} vers une conjonction de prédicats de Σ_{HQ}. En réécrivant la preuve du théorème de l'orthocentre dans Σ_{HQ} nous constatons que la taille des formules a bien augmenté mais que le nombre de règles à appliquer pour arriver au résultat final a fortement baissé :

	Σ_{CGZ}	Σ_{HQ}
nb. de prédicats pour les hypothèses	bas (9)	élevé (59)
nb. d'applications de règles	élevé (68)	bas (7)
nb. de règles nécessitées	élevé (17)	bas (4)
taille moyenne des règles	bas (2,29)	élevé (21,5)
nb. de types utilisé	bas (1)	élevé (6)

Nous avons donc baissé le nombre de règles d'inférence à appliquer et augmenté l'intelligibilité des formules par le choix de types intuitifs, au dépens de la taille des formules (qui n'utilisent que des prédicats binaires, ternaires et quaternaires, contrairement aux prédicats d'arité variable de CGZ).

Il ne reste donc qu'à pallier le problème de la taille des formules en les présentant de manière synthétique. Pour cela nous nous sommes tournés vers les *graphes conceptuels variadiques*.

2.1.3 Graphes conceptuels de base/simples

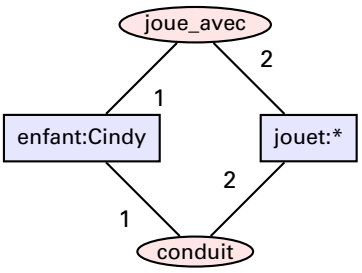
Les *graphes conceptuels de base* (BG) et les *graphes conceptuels simples* (SG) sont définis dans [Chein et Mugnier, 2009] de la manière suivante :

Soit T_C un ensemble partiellement ordonné, appelé ensemble des *types de concept* ; T_R un ensemble partiellement ordonné, appelé ensemble des *types de relation* ; et $\mathcal{I}$ l'ensemble des *référents individuels* auquel on ajoute l'élément $\{*\}$, appelé *référent générique*. On appelle le triplet $\mathcal{V} = (T_C, T_R, \mathcal{I})$ un *vocabulaire*.

Un *graphe conceptuel de base* défini sur un vocabulaire $\mathcal{V}$ est la donnée d'un quadruplet $G = (C, R, E, \ell)$ où C (concepts) et R (relations) sont les partitions d'un graphe biparti et E ses arêtes. L'application ℓ (label) attribue à chaque concept une paire (type, référent) $\in T_C \times (\mathcal{I} \cup \{*\})$ et à chaque relation un élément de T_R. Les arêtes d'une relation r sont numérotées de 1 à arité(r).

Pour illustrer ces notions prenons un exemple. La photo ci-contre, tirée d'une banque d'images commerciales, peut être annotée par la phrase « Un enfant joue en conduisant une voiture-jouet »[3]. Nous pouvons choisir d'utiliser les types de concept $T_C = \{\mathsf{jouet}, \mathsf{enfant}\}$ et les types de relation $T_R = \{\mathsf{joue\_avec}, \mathsf{conduit}\}$, toutes deux d'arité 2. Notre graphe aura deux instances de concept jouet :* et enfant :*, dans les deux cas le référent est générique. Si on sait que l'enfant s'appelle Cindy, et la phrase devient donc « Cindy joue en conduisant une voiture-jouet » et l'instance de concept de l'enfant sera enfant :Cindy. Voici le graphe conceptuel de base en question :

[3]La photo est tirée du site 123RF (son identifiant est 81689702) et sa description officielle, bien plus complète, est « Petite fille vêtue de chemise et de chapeau noirs de points noirs, conduisant une voiture de jouet rouge près du parc. Bébé fille, enfant qui joue avec la voiture ». Cet exemple n'est pas anodin : une recherche « intelligente » dans une base d'images est une application paradigmatique des graphes conceptuels.



À cela on peut remarquer que le jouet est une voiture, on a donc deux types de concept : « jouet » et « voiture » qui sont incomparables, dans le sens que l'un n'est pas hyponyme de l'autre. Cela est pris en compte par la définition de *type conjonctif* :

Soit $\{t_1, \ldots, t_n\}$ un ensemble de types incomparables deux-par-deux. On définit un *type conjonctif* $t_1 \wedge \cdots \wedge t_n$ comme un tel ensemble, doté d'un ordre partiel hérité de l'ordre de T_C : si $t = t_1 \wedge \cdots \wedge t_n$ et $t' = t'_1 \wedge \cdots \wedge t'_m$ sont deux types conjonctifs alors $t \leq t'$ ssi pour tout t'_i, $1 \leq i \leq m$ il existe un t_j, $1 \leq j \leq n$ tel que $t_j \leq t'_i$.

Pour un ensemble de types T, l'ensemble de type conjonctifs $(T^{\sqcap}, \leq)$ est un treillis. On définit un ensemble $\mathcal{B} \subset T^{\sqcap}$ appelé *ensemble des types bannis de base* et $\mathcal{B}^*$ l'union des idéaux générés par $\mathcal{B}$, appelé *ensemble des types bannis*.

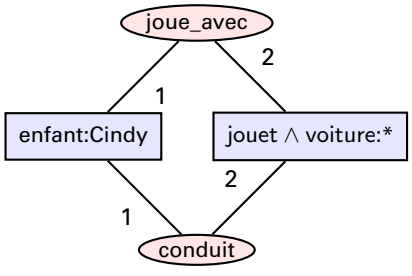
Un *vocabulaire conjonctif* est un triplet $(T_C(T, \mathcal{B}), T_R, \mathcal{I})$ où $T_C(T, \mathcal{B}) := T^{\sqcap} \setminus \mathcal{B}^*$. Les types bannis nous permettent d'éviter la combinaison de types incompatibles. Cela nous permet de définir des ensembles de *concepts compatibles* : ce sont des concepts qui ont au plus un référent individuel et dont le type conjonctif minimal appartient à T_C (donc : n'est pas banni).

Pour reprendre notre exemple, on pourrait utiliser dans le graphe conceptuel de description de la photo le type conjonctif voiture $\wedge$ jouet. Il n'est pas banni puisqu'il n'y a pas d'incompatibilité de principe entre les concepts de voiture et de jouet (depuis des temps immémoriaux les enfants jouent avec des miniatures de voitures), un exemple de types incompatibles serait, par exemple si on ajoutait la relation éprouve et le type de concept joie (qui est un sentiment), la conjonction voiture $\wedge$ joie.

La notion de concepts compatibles permet de définir une relation d'équivalence de *coréférence* : on définit des sous-ensembles de T_C appelés *classes de coréférence* qui sont des ensembles de concepts compatibles tels que deux concepts avec le même référent individuel appartiennent à la même classe. Autrement dit : une classe de coréférence peut ne pas avoir de concept avec référent individuel, mais pour chaque référent individuel il n'existe qu'une classe contenant tous les concepts possédant ce label et aucun concept possédant un autre référent individuel.

Le quintuplet $(C, R, E, \ell, \mathrm{coref})$, où (C, R, E, ℓ) est un BG et coref une relation d'équivalence de coréférence, est un *graphe conceptuel simple*.

Pour reprendre une dernière fois l'exemple de la photo, comme ici il n'y a qu'un seul enfant et une seule voiture la relation d'équivalence de coréférence ne change pas la structure du graphe. Le graphe



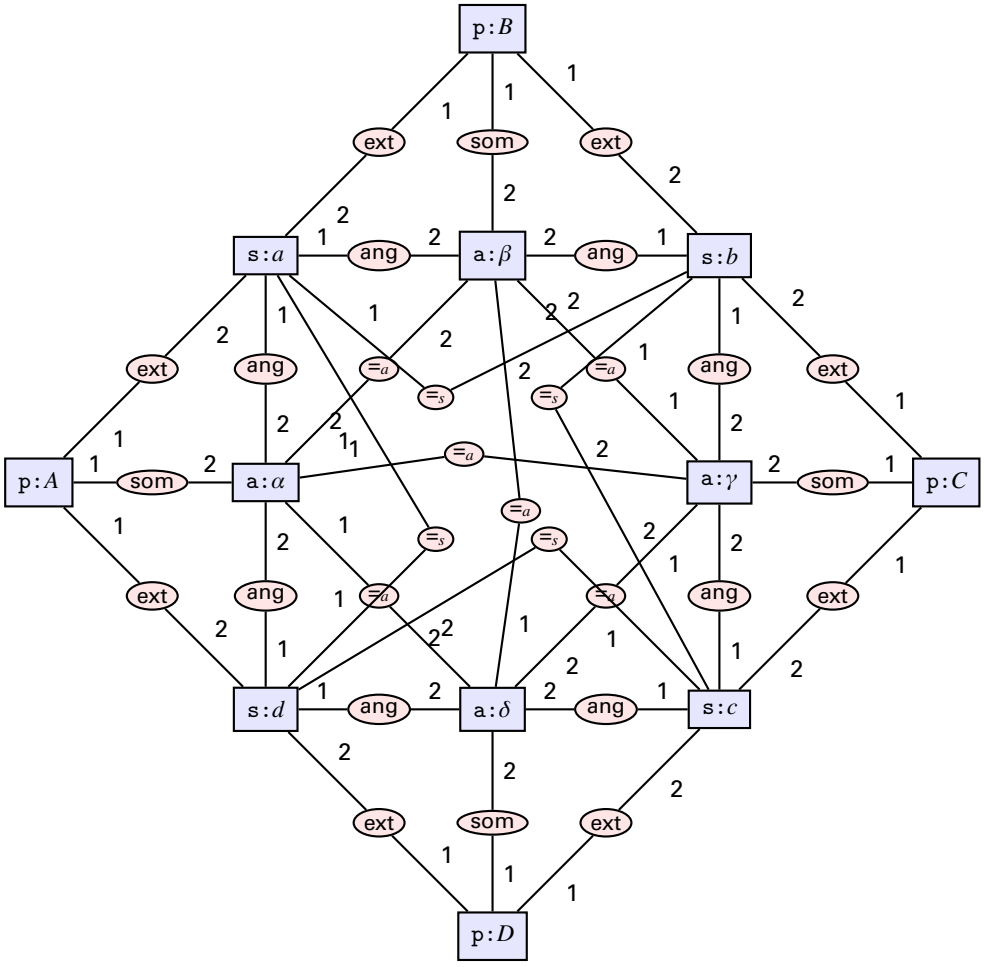
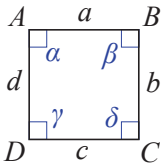


FIG. 2.1: Graphe conceptuel du carré *ABCD* de côtés a, b, c, d et d'angles α, β, γ, δ. Parmi les relations d'égalité d'angle nous n'avons représenté que 6 sur 12, et parmi les relations d'égalité de côté, 5 sur 12.

où les rectangles dénotent non pas des instances de concept mais des instances de classe de coréférence de concepts, est un graphe conceptuel simple.

Pour revenir aux figures géométriques, prenons le cas du carré *ABCD* de côtés a, b, c, d et d'angles α, β, γ, δ :



Nous présentons sur la fig. 2.1 le graphe correspondant. Les relations sont ext (un point est extrémité d'un segment), ang (un segment fait partie d'un angle), som (un point est sommet d'un angle), $=_a$ (égalité d'angles), $=_s$ (égalité de longueurs de segments). Toutes sont des relations binaires. Nous ne les avons pas toutes représentées pour ne pas surcharger la figure.

Un résultat important de Chein et Mugnier [2009] est *qu'il existe une transformation* Φ *qui en-*

voie un graphe conceptuel simple vers une conjonction de formules logiques conjonctives existentielles sans fonctions, telle que pour deux graphes conceptuels simples G et H sur le même vocabulaire $\mathcal{V}$ la conséquence $\Phi(\mathcal{V}) \wedge \Phi(H) \models \Phi(G)$ est équivalente à l'existence d'un homomorphisme de graphes conceptuels simples $G \to H$.

Ce résultat est important parce qu'il ramène la preuve d'une conséquence à la recherche d'un homomorphisme entre graphes. La transformation Φ peut être décrite comme suit :

- pour toute paire de types de concept primitifs t_1 et t_2 tels que $t_2 < t_1$, $\Phi(\mathcal{V})$ contient une formule $\forall x\, t_2(x) \to t_1(x)$;
- pour toute paire de types de relation t_1 et t_2 tels que $t_2 < t_1$ (ils sont alors de même arité k), $\Phi(\mathcal{V})$ contient une formule $\forall x_1 \cdots \forall x_k\, t_2(x_1, \ldots, x_k) \to t_1(x_1, \ldots, x_k)$;
- on associe à chaque classe de coréférence contenant un référent individuel i une constante i ;
- on associe à chaque classe de coréférence ne contenant pas de référent individuel une variable ;
- on associe à chaque concept c appartenant à une classe de coréférence la constante ou la variable correspondante, notons cela $\mathrm{term}(c)$;
- Φ associe à chaque concept c de type $t_1 \wedge \cdots \wedge t_n$ la formule $t_1(\mathrm{term}(c)) \wedge \cdots \wedge t_n(\mathrm{term}(c))$;
- Φ associe à chaque relation r d'arité k le prédicat $r(\mathrm{term}(c_1), \ldots, \mathrm{term}(c_k))$ où les c_i correspondent aux concepts voisins de r ;
- $\Phi(G)$ est la clôture existentielle de la conjonction des formules associées à tous les concepts et relations de G.

2.1.4 Graphes conceptuels variadiques

Nous introduisons [YH et Quaresma, 2018a] une extension des graphes conceptuels simples appelée *graphes conceptuels variadiques*[4] Celle-ci est adaptée à la formalisation de la géométrie euclidienne, elle consiste *grosso modo* à autoriser des relations d'arité variable et avec des arêtes non-ordonnées.

Les relations qui nous intéressent ont ceci de particulier qu'elles sont symétriques et dotées d'une *arité minimale naturelle*. Par exemple, il faut au moins trois points pour parler de colinéarité et au moins quatre points pour parle de cocyclicité, ce sont les arités minimales de ces deux relations ; d'autre part la colinéarité et la cocyclicité sont des relations symétriques : l'ordre de leurs arguments n'a aucune importance. Nous définissons donc :

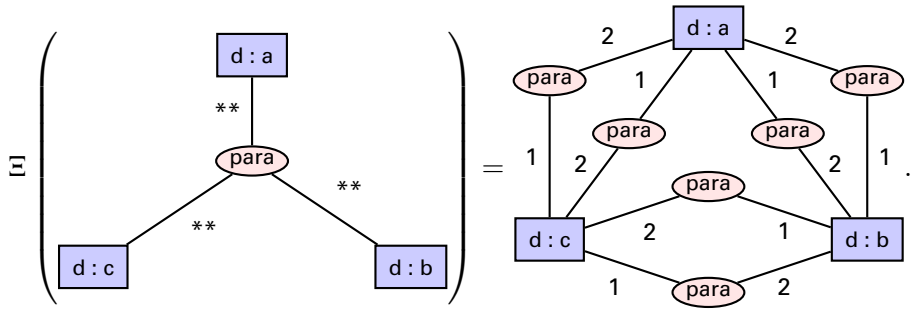
1. une *multi-arité* de type de relation : il s'agit d'une paire de valeurs (n, m) où
   - n dénote le nombre d'arêtes « traditionnelles » de la relation, c'est-à-dire d'arêtes numérotées et de nombre constant dans toutes les instances, et
   - m dénote l'« arité minimale » du type de relation, qui dépend de la nature de la relation.

   À noter que, contrairement aux graphes conceptuels traditionnels, l'arité K d'une instance de relation est *variable*, la seule contrainte posée étant que $K \geq n + m$;

[4] Le terme « variadique » provient d'un exposé du grand philosophe américain Nelson Goodman et de son collègue Henry S. Leonard, intitulé *A calculus of individuals* et donné à la 2e réunion de la celèbre *Association for Symbolic Logic*, à Harvard, entre Noël 1936 et le Nouvel An 1937 [Leonard et Goodman, 1937]. Il est intéressant de noter que dans la toute première utilisation de ce terme, l'exemple donné relève de la géométrie : *One great extrinsic value of the calculus lies in its making possible the analysis of "variadic" relations, such as, e.g., "lying together in a straight row," for which the number of arguments occurring in the various values is not a constant.* La réunion a été suivie par un thé au Harvard Faculty Club pour les membres de l'Association et leurs amis. À noter que quand Goodman & Leonard publièrent leur calcul des invididus dans la même revue trois ans plus tard, ils avaient remplacé le terme « variadique » par « multigrade », cf. Leonard et Goodman [1940, p. 50]. Concernant la logique variadique, voir aussi Alexander [2013].

2. un label « $**$ » affecté aux arêtes non-ordonnées de relation ;
3. une application Ξ (appelée *expansion*) qui envoie tout graphe conceptuel variadique vers un graphe conceptuel simple, et son inverse Ξ^{-1} (appelée *contraction*) qui envoie tout graphe conceptuel simple vers un graphe conceptuel variadique.

Voici un exemple : pour dire que 3 droites a, b, c sont parallèles, on utilise en logique du 1er ordre une conjonction de 6 prédicats binaires : $P(a,b) \wedge P(a,c) \wedge P(b,a) \wedge P(b,c) \wedge P(c,a) \wedge P(c,b)$. Dans ce cas, la multi-arité du prédicat de parallélisme est $(0,2)$ (aucune arête ordonnée, et il faut au moins deux droites pour parler de parallélisme). Comme il s'agit de trois droites, le degré total est $K = 3$ et donc le nombre de relations `para` obtenues dans l'expansion du graphe est de $\frac{(K-n)!}{(K-m-n)!} = \frac{3!}{1!} = 6$.



D'autre part, sur la fig. 2.2 nous représentons le graphe conceptuel variadique qui est la contraction du graphe de la fig. 2.1. Il n'y a que deux relations d'égalité (une pour les segments et une pour les angles). Toutes les relations sont représentées sur la figure. On observe 24 arêtes ordonnées (pour 12 relations de multi-arité $(2,0)$) et 24 arêtes non-ordonnées (pour 8 relations « ang » de multi-arité $(0,2)$ et d'arité effective 2, et 2 relations d'égalité de multi-arité $(0,2)$ et d'arité effective 4).

Nous montrons dans [YH et Quaresma, 2018a] que si $\phi : G \to H$ est un homomorphisme de graphes conceptuels variadiques alors $\Xi(\phi)$ est un homomorphisme de graphes conceptuels simples, et inversement si $\phi' : G' \to H'$ est un homomorphisme de graphes conceptuels simples il existe $\Xi^{-1}(\phi')$ qui est homomorphisme de graphes conceptuels variadiques.

En combinant avec le théorème [Chein et Mugnier, 2009, Theorem 4.3] nous obtenons donc le fait que l'existence d'un homomorphisme de graphes conceptuels variadiques équivaut à la conséquence des formules associées par $\Phi \circ \Xi$ en logique du 1er ordre.

Nous définissons des graphes conceptuels variadiques qui correspondent à la signature Σ_{HQ}. Les règles déductives de HQ deviennent des règles de ré-écriture de graphe conceptuel variadique.

2.1.5 Fouille de figures géométriques

Depuis quelques années on assiste à l'émergence d'entrepôts numériques contenant des milliers de figures géométriques [Quaresma, 2011], ce qui nous a conduit à nous poser la question de la fouille de figures géométriques, la difficulté étant que plusieurs propriétés sont équivalentes et leur détection devrait être indépendante de la manière dont la requête est posée. Ainsi, par exemple, deux droites sont parallèles quand (1) l'angle qu'elles forment est nul (aux conditions de non-dégénérescence près) ou (2) quand leur distance minimale est constante ou (3) quand elles intersectent une autre droite avec les mêmes angles, etc. Si la propriété est représentée d'une des trois manières dans le motif de recherche et par une autre dans la figure du corpus, cette dernière ne sera pas détectée.

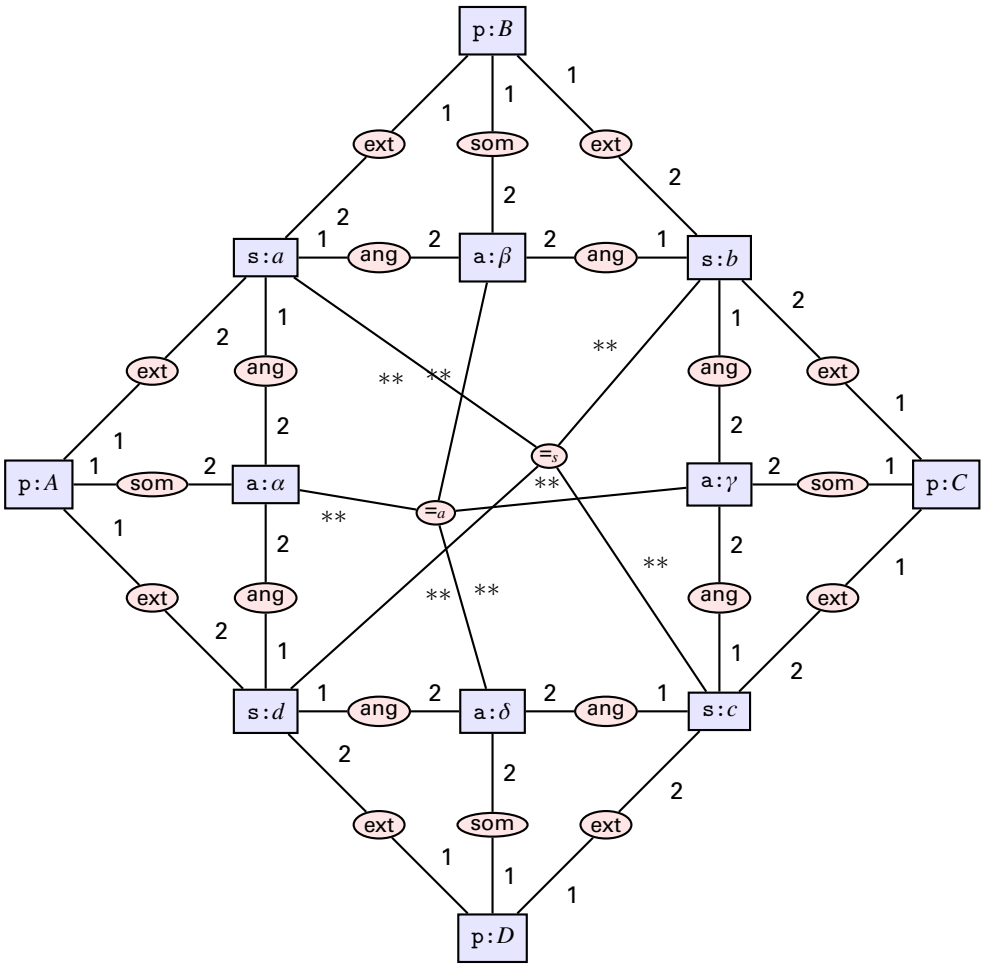


FIG. 2.2: Graphe conceptuel variadique du carré *ABCD* de côtés a, b, c, d et d'angles α, β, γ, δ. Toutes les relations sont représentées.

Dans l'univers de CGZ, le passage d'une représentation de propriété donnée à une autre se fait par des déductions. Théoriquement cela est toujours possible, mais on se heurte à des problèmes de performance.

En effet, la recherche de figures répondant à des critères spécifiques dans un corpus volumineux de figures géométriques nous impose trois contraintes majeures :

1. il faut limiter au maximum l'utilisation du moteur d'inférences,
2. on doit être capable d'éliminer très rapidement les figures incompatibles,
3. si aucune figure répondant aux critères stricts n'est trouvée, on doit être capable d'affaiblir les contraintes pour trouver des réponses approximatives.

Dans [YH et Quaresma, 2018b] nous décrivons notre solution à ces trois points. Concernant le point 1 nous effectuons des clôtures inférentielles du corpus en amont et la clôture inférentielle de la requête dès que celle-ci est posée. Pour cela nous appliquons toutes les règles d'inférence en boucle jusqu'à atteindre un point fixe.

La détection de motifs de graphes pour appliquer des règles déductives se fait normalement à travers une implémentation de l'algorithme d'isomorphisme de sous-graphe [Peixoto, 2014], mais celle-ci est gourmande en temps et en CPU. Pour accélérer cette détection et résoudre, en même temps, le point 2, nous avons établi une *vérification de distribution de sentiers*. Dans un graphe, un sentier est un chemin qui ne repasse pas par la même arête. Dans notre cas, une *distribution de sentiers* est une distribution de chaînes de symboles correspondant aux types d'arêtes des sentiers du graphe. Si on veut qu'un graphe de requête soit inclus dans un graphe du corpus, alors ce dernier doit au moins posséder une distribution égale ou supérieure à celle du graphe de requête, et cela est très facile à vérifier. Une fois cette étape passée, on applique l'algorithme d'isomorphisme de sous-graphe, qui permet de valider ou invalider la présence du graphe de requête dans le graphe cible et, le cas échéant, permet d'obtenir toutes ses occurrences dans le graphe. Si la vérification échoue, le graphe est éliminé.

Pour le point 3 nous réitérons la vérification de distribution de sentier pour une suite de motifs réduits, auxquels on enlève successivement des nœuds de relation. La vitesse élevée à laquelle nous effectuons la vérification de distribution de sentiers nous permet d'obtenir rapidement une série de sous-motifs maximaux qui apparaissent effectivement dans le graphe cible.

2.2 Aide à la navigation maritime

Les travaux qui suivent ont été menés dans le cadre d'un projet de recherche financé par le Service hydrographique et océanographique de la Marine, dont le but était de considérer l'utilisation de cartes électroniques maritimes en tant qu'objets riches, interactifs et mis-à-jour par les utilisateurs. Les cartes maritimes du Shom sont, depuis 1720 (publiées à l'époque par l'ancêtre du Shom, le Dépôt des cartes et plans de la Marine), une véritable institution pour les navigants, que ce soit en eaux françaises territoriales ou internationales. Conscient du fait qu'une carte ne peut contenir que certains types d'informations (principalement géographiques, et dans une moindre mesure, administratives et historiques), le Shom a très tôt fait accompagner ses cartes d'une séries d'ouvrages appelés *Instructions nautiques* [Menanteau, 2013, 2011], dont la possession est obligatoire pour les navires marchands et militaires. Et une fois cette multimodalité communicative établie, le Shom n'a cessé d'enrichir ces deux vecteurs de communication : les cartes (papier) et les ouvrages les accompagnant.

À l'heure de la numérisation des cartes et des dispositifs de navigation électroniques on s'est proposé d'intégrer dans une même ressource, une base de connaissances accessible en ligne, aussi bien les informations traditionnellement contenues dans les cartes que les informations (et connaissances) contenues dans les *Instructions nautiques*. Ces informations étant de modalités différentes (textuelle et graphique) tout en étant étroitement liées, nous avons défini un nouveau type de langage contrôlé : le *langage contrôlé hybride*, doté de deux grammaires formelles, deux arbres syntaxiques mais une représentation sémantique commune.

2.2.1 La composante visuelle

Pour incorporer dans le langage hybride les informations (et connaissances) provenant des cartes nous nous sommes servis des *grammaires SR* introduites en 1996 par Ferrucci et son équipe [Ferrucci et al., 1996, 1998]. Les grammaires SR (ou SR est l'acronyme de « symboles-relations ») sont adaptées à tout langage formel nécessitant un nombre élevé de concaténateurs.

L'idée est la suivante : traditionnellement, dans un langage formel « à la Chomsky », c'est-à-dire

avec un seul concaténateur, on utilise des symboles pour les membres de l'alphabet et un mot s'écrit comme une suite de ces symboles. Par exemple, dans un langage basé sur l'alphabet $\{a, b\}$, les mots s'écrivent comme *ab*, *abba*, *bbb*, ε (le mot vide), etc. Dans ce que nous venons de dire, deux choses sont passées sous silence :

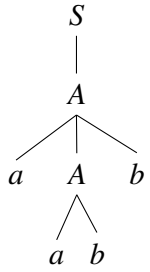
1. comme il n'y a qu'un seul concaténateur, on ne se donne pas la peine de le dénoter par un graphème ;
2. comme l'ordre des symboles est significatif, on peut écrire plusieurs fois le même symbole et il sera identifié par son ordre dans le mot : ainsi on peut parler sans ambiguïté du «2e *b* du mot *abba*».

Lorsque les membres de l'alphabet sont des éléments graphiques que l'on dispose dans l'espace selon certaines règles, on n'a plus d'ordre linéaire. Ainsi, par exemple, dans les schémas de génie électrique les composants sont reliés par des lignes qui représentent les chemins de transmission : les composants sont alors les membres de l'alphabet et les différentes manières de les connecter entre eux sont les concaténateurs. De ce fait les deux propriétés des langages «à la Chomsky» énoncées ci-dessus deviennent caduques : un mot n'est plus qu'un ensemble (non-ordonné), et pour distinguer les instances de membres de cet ensemble qui apparaissent dans les mots, il faut les indexer.

Pour illustrer le mécanisme des grammaires SR prenons l'exemple du langage $\{\underbrace{a \ldots a}_{n \text{ fois}} \underbrace{b \ldots b}_{n \text{ fois}}\}$ pour $n \geq 1$, exemple paradigmatique de langage non-régulier, malgré sa simplicité. Dans le formalisme de Chomsky, une grammaire possible pour ce langage possède deux terminaux $\{a, b\}$, deux non-terminaux $\{S, A\}$ et les règles de production suivantes :

$$\begin{aligned} s_0 &: \quad S \to A, \\ s_1 &: \quad A \to aAb, \\ s_2 &: \quad A \to ab. \end{aligned}$$

Ce qui nous donne l'arbre syntaxique suivant pour le mot *aabb* :



Selon le formalisme SR, le même mot *aabb* s'écrira comme une paire d'ensembles

$$<\{a^1, a^2, b^2, b^1\}, \{\mathsf{next}(a^1, a^2), \mathsf{next}(a^2, b^2), \mathsf{next}(b^2, b^1)\}>, \tag{2.1}$$

où le premier ensemble contient les symboles participant au mot et le deuxième contient les relations entre ces symboles. Ici, next est la relation qui représente le seul et unique concaténateur dont nous avons besoin, et les exposants sont des labels permettant d'identifier les instances de symboles.

Une *grammaire SR* contient deux types de règles de production :

- les *s-règles* s'appliquent aux symboles non-terminaux et produisent des paires d'ensembles (symboles et relations) ;

- les *r-règles* s'appliquent aux relations et produisent des ensembles de relations.

On applique d'abord une s-règle, et ensuite on a le droit d'appliquer des r-règles (chaque r-règle possède dans sa définition l'ensemble des s-règles qui la rendent disponible).

Une convention très déroutante lorsqu'on aborde pour la première fois les grammaires SR est le fait que lorsqu'on écrit un symbole non-terminal en tant que prémisse de la règle de production et que ce symbole ré-apparaît dans la conclusion de la règle, on considère que c'est une *autre* instance du même symbole et *on lui affecte un label différent*. Avec une complication supplémentaire : lorsqu'il s'agit d'une s-règle, on ne va incrémenter le label du non-terminal que dans l'ensemble des symboles, et *c'est une r-règle qui va changer le label du non-terminal dans la partie relations*. Illustrons cela par l'exemple.

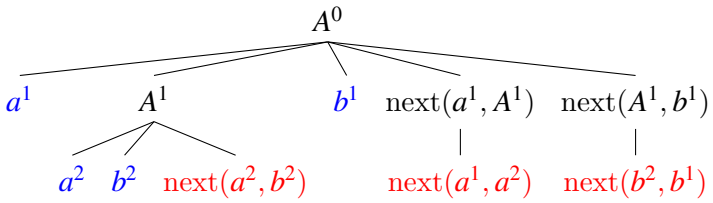
Voici les s-règles pour le langage $\{\underbrace{a \dots a}_{n \text{ fois}} \underbrace{b \dots b}_{n \text{ fois}}\}$:

$$
\begin{aligned}
s_0 : &\quad S \to <\{A^1\}, \emptyset> \\
s_1 : &\quad A^1 \to <\{a^1, A^2, b^1\}, \{\text{next}(a^1, A^1), \text{next}(A^1, b^1)\}> \\
s_2 : &\quad A^1 \to <\{a^1, b^1\}, \{\text{next}(a^1, b^1)\}>.
\end{aligned}
$$

Pour nous débarrasser du A^1 qui continue d'exister dans les relations de s_1 nous introduisons deux r-règles :

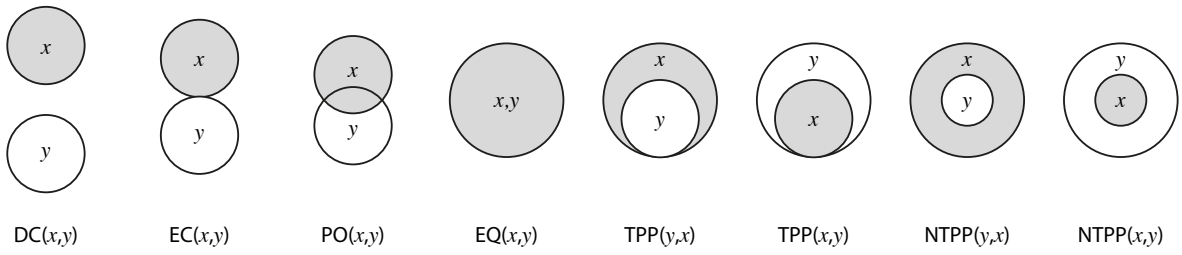
$$
\begin{aligned}
r_{\{1\},1} : &\quad \text{next}(*, A^1) \to \{\text{next}(*, A^2)\} \\
r_{\{1\},2} : &\quad \text{next}(A^1, *) \to \{\text{next}(A^2, *)\},
\end{aligned}
$$

où $*$ peut être n'importe quel terminal ou non-terminal (il y a donc autant de paires de règles que de symboles terminaux et non-terminaux). Le premier indice de $r_{\{1\},1}$ et de $r_{\{1\},2}$ indique que ces r-règles peuvent être appliquées après une application de la règle s_1. En appliquant bien s_0 à S, puis s_1, puis $r_{\{1\},1}$ et $r_{\{1\},2}$, et enfin s_2, on obtient bien le mot (2.1). Voici l'arbre syntaxique de cette dérivation :



où nous avons colorié en bleu les terminaux appartenant à l'ensemble des symboles et en rouge les terminaux appartenant à l'ensemble des relations (rappelons qu'il s'agit bien d'ensembles non-ordonnés et que seule l'existence d'une relation « next » entre deux instances de symbole permet de déduire leur « ordre » dans le mot).

Pour représenter les cartes en utilisant ce formalisme nous prenons les objects géographiques comme symboles et les relations de la logique topologique RCC8 de Randell et al. [1992] comme relations :



2.2.2 Un énoncé hybride

Les informations graphiques et textuelles sont étroitement liées : elles sont complémentaires et souvent co-référentielles. Voici un exemple d'énoncé hybride (un énoncé graphique et un énoncé textuel, liés) très simple :

[L'épave du Morania 130] gît au fond du [lake Erie]. (2.2)

L'extrait de carte comporte deux objets visibles :  et  , ainsi que leurs légendes. Dans l'énoncé textuel nous avons deux entités nommées : «[lake Erie]» et «[l'épave du Morania 130]». Ici, «  » et «[lake Erie]» sont des entités coréférentielles : leur référent commun dans la réalité est le lac Érié (entre le Canada et les US). De même, «  » et «[l'épave du Morania 130]» sont des co-références à l'épave du Morania #130, un bateau qui a naufragé en 1951 et dont l'épave gît au fond du lac Érié.

Pour décrire les relations RCC8 des objets graphiques de l'extrait de carte nous allons considérer les figures atomiques  et  et leur légendes en tant que symboles de terminaux SR. Une relation, que nous nommons λ («lexical»), fait le lien entre graphique et légende. La syntaxe de l'énoncé visuel est représenté en bas à gauche de la fig. 2.3. À cet arbre nous pouvons également ajouter de l'information provenant de la carte électronique mais qui n'est pas nécessairement visible, comme le type d'objet (ici nous avons deux types : `Wreck` (épave) et `Lake` (lac)), et les coordonnées géographiques. En extrayant le sens de l'arbre syntaxique nous obtenons une formule de logique du 1er ordre comme suit :

$$\mathsf{Wreck}(\text{}) \wedge \mathsf{Lake}(\text{  }) \wedge \mathsf{NTPP}(\text{}, \text{}) \wedge \lambda(\text{}, \text{«Lake Erie»}) \wedge \lambda(\text{}, \text{«Morania 130»}). \quad (2.3)$$

Du côté textuel, l'énoncé «[L'épave du Morania 130] gît au fond du [lake Erie]» donne lieu à l'arbre syntaxique de la partie droite de la fig. 2.3. Si nous appelons M et E le constantes logiques qui correspondent aux entités géolocalisées «[L'épave du Morania 130]» et «[lake Erie]», alors la sémantique de l'énoncé textuel peut être représentée comme suit :

$$\mathsf{lies}(\mathsf{M}, \mathsf{bottom}(\mathsf{E})) \quad (2.4)$$

où lies est un prédicat binaire et bottom une fonction unaire.

Après une étape de coréférence qui va fusionner  et E, ainsi que  et M, nous pouvons fusionner les formules logiques (2.3) et (2.4) en une seule, représentée en haut de la fig. 2.3.

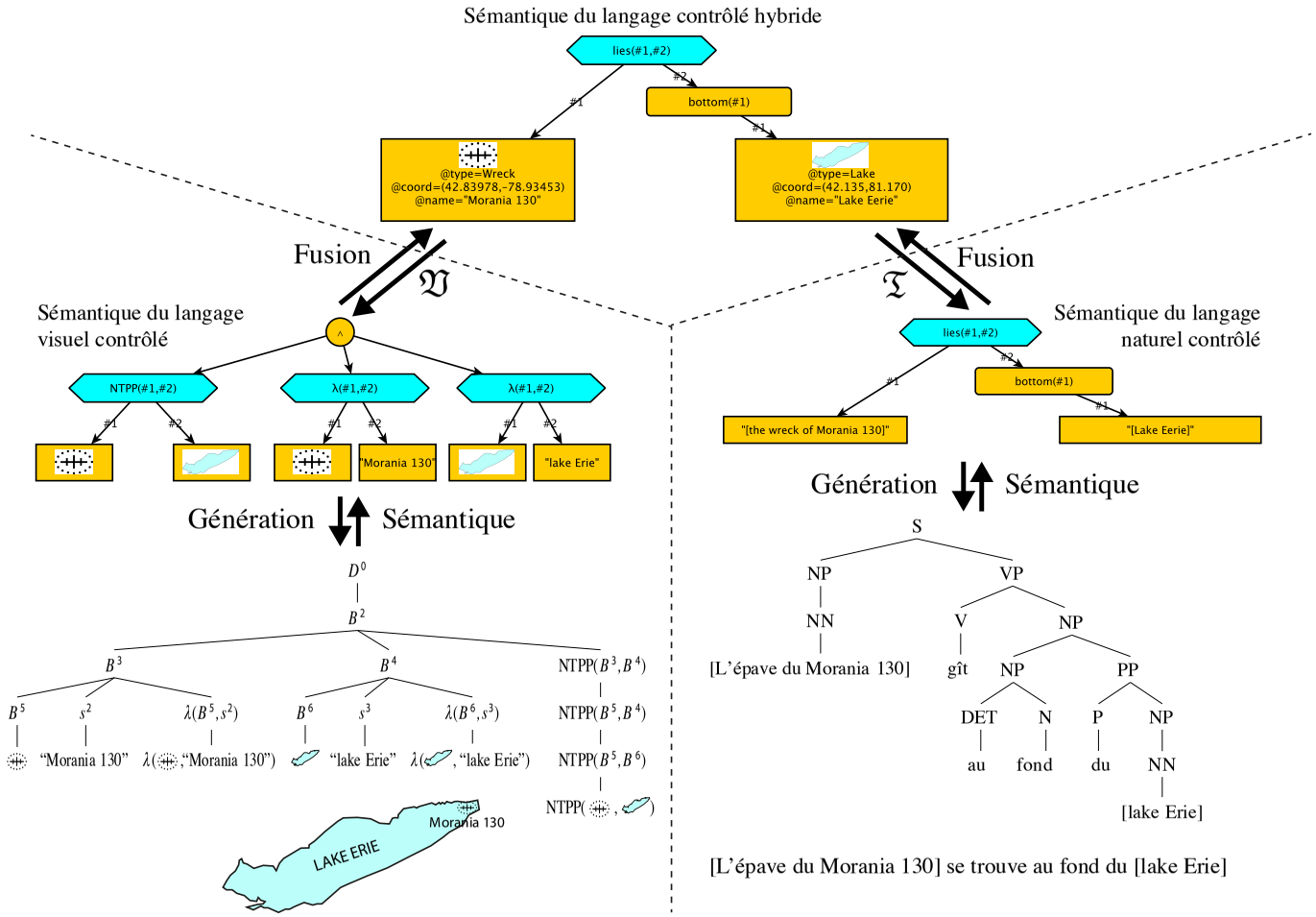


FIG. 2.3: Arbres syntaxiques visuel (à gauche) et textuel (à droite), et représentation sémantique commune (en haut) de l'énoncé hybride (2.2).

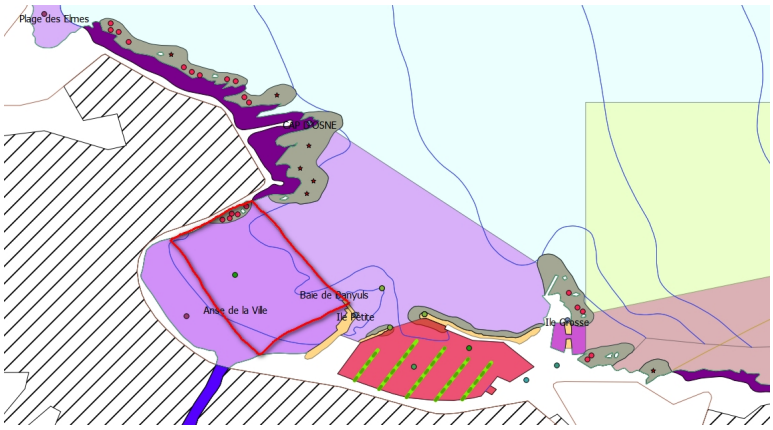


FIG. 2.4: Fragment de carte électronique.

Comme on peut le voir dans cet exemple, chaque modalité porte de l'information spécifique : l'énoncé visuel fournit l'emplacement des objets, leurs tailles et formes. Leur complémentarité apparaît clairement dans le cas de l'épave : l'énoncé textuel précise qu'elle gît « au fond du lac », information que la carte 2D ne peut afficher, alors que l'énoncé visuel indique l'emplacement de l'épave en utilisant le symbole qui a la sémantique « épave submergée », une information moins précise que celle de l'énoncé textuel.

2.2.3 INAUT, un langage pour les *Instructions nautiques*

Dans [YH et al., 2017] nous définissons un langage hybride destiné à alimenter la base de connaissances du Shom, nous l'appelons INAUT. La raison d'être d'INAUT est de transmettre aussi bien des informations textuelles que des informations visuelles provenant des cartes (considérées comme objets informatiques). Nous présentons ci-dessous un exemple de paragraphe écrit dans la modalité textuelle d'INAUT. Ce texte provient des *Instructions nautiques* (Vol. 2.1 § 2.2.4), après avoir subi une détection et un géoréférencement des entités nommées et une résolution d'anaphores :

> La [baie de Banyuls] est limitée au NW par le [cap d'Osne]. La [baie de Banyuls] est limitée à l'Est par l'[île Grosse]. L'[île Grosse] est rattachée à la côte par un terre-plein. La [baie de Banyuls] est divisée en deux parties par l'[île Petite] : à l'Ouest l'[anse de la Ville]et à l'Est l'[anse de Fontaulé]. L'[île Petite] est reliée au rivage par un terre-plein. L'[anse de la Ville] est bordée par une plage. La plage est dominée par l'agglomération. L'[anse de Fontaulé] abrite le port. La [baie de Banyuls] possède deux mouillages. Le premier mouillage est localisé au NE du [Cap d'Osne]. Le premier mouillage a une profondeur de 20 mètres. Le premier mouillage est de type sable et gravier. Le premier mouillage a une mauvaise tenue. Le deuxième mouillage est localisé à l'ouvert de l'[Anse de la Ville]. Le deuxième mouillage a une profondeur de 5 à 6 mètres. Le deuxième mouillage est protégé des vents de Nord. Le deuxième mouillage est protégé des vents de Nord-Ouest. Le deuxième mouillage est intenable par vents d'Est.

Le fragment (visuel) afférent de carte électronique est représenté sur la fig. 2.4. Sur la fig. 2.5 nous représentons sous forme de graphe conceptuel le fragment de base de connaissance qui correspond à l'extrait de langage hybride dont la composante textuelle est l'extrait ci-dessus et la composante visuelle une partie du fragment de la fig. 2.4. Dans ce graphe les concepts et relations provenant de la composante textuelle sont en bleu, ceux provenant de la carte en rose, ceux provenant aussi bien du texte que de la carte en orange, et enfin les concepts en jaune (nous n'avons pas représenté la relation

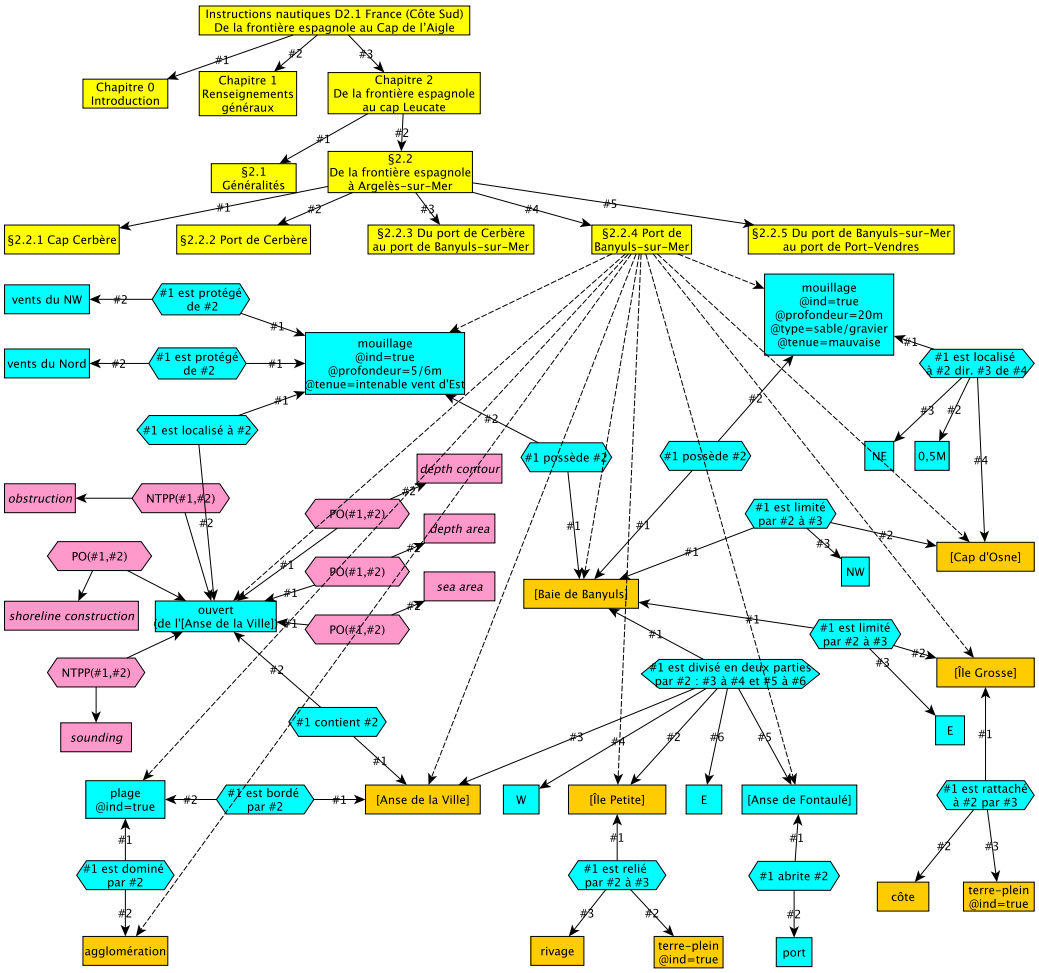


FIG. 2.5: Représentation sous forme de graphe conceptuel d'un fragment de base de connaissances.

d'imbrication hiérarchique) représentent les subdivisions hiérarchiques du volume correspondant des *Instructions nautiques*. Les flèches en pointillé indiquent l'occurrence d'une entité nommée dans une subdivision hiérarchique donnée.

2.2.4 Génération de texte

Le langage INAUT doit remplir deux tâches : permettre à un rédacteur ou un navigateur (de confiance) de soumettre des modifications à la base de connaissances, mais aussi générer du texte à partir de fragments de la base de connaissance. En effet, la carte électronique peut afficher certaines couches d'information visuelle mais la transmission d'une grande partie des informations contenues à l'origine dans les *Instructions nautiques* (informations administratives, réglementaires, culturelles, historiques, etc.), nécessite le passage par la modalité textuelle (à travers des fenêtres de texte qui s'affichent près de la zone concernée).

Nous nous sommes donc penchés, dans [Sauvage-Vincent et al., 2015] sur la *génération de texte* à partir de fragments de la base de connaissances. Obtenir un énoncé textuel pour chaque relation de la base de connaissances ne présente aucune difficulté particulière, après tout la base a été alimentée

au départ à partir de données textuelles dans une langue de spécialité bien structurée et avec un minimum de polysémie.

Le véritable problème que nous avons rencontré est l'ordonnancement des phrases correspondant aux relations. Lorsque l'utilisateur visionne une zone de la carte, celle-ci correspond à un sous-graphe de la base de connaissances et la génération de texte commence par le choix des concepts et relations à utiliser, qui est relativement facile. Mais ces concepts et relations n'ont aucun ordre intrinsèque, alors que dans un paragraphe de texte l'ordre des phrases est d'une importance capitale.

Sur un corpus de 426 extraits des *Instructions nautiques* nous avons étudié les choix d'ordre de description des différents éléments de la carte. Voici les principaux facteurs émergeant qui déterminent ces choix :

- les repères (naturels ou artificiels) : les rédacteurs des *Instructions nautiques*, anciens navigateurs pour la plupart, se sont avant tout basés sur les repères visibles à un navigateur à partir de la position donnée, ce qui concorde avec Michon et Denis [2001] ;
- les primitives géométriques : si les différents objets contenus dans une carte correspondent topologiquement à des aires, des lignes et des points, on a constaté que les rédacteurs ont tendance à les décrire dans cet ordre, ce qui est confirmé par Brosset et al. [2008] qui affirme qu'un observateur dans un environnement naturel perçoit son environnement sous forme de réseau spatial, dans lequel l'importance est accordée aux objets par ordre décroissant du nombre de dimensions ;
- le nom et la taille : les objets nommés ainsi que ceux de grande taille sont décrits avant les objets anonymes ou petits ;
- les espaces de proximité : selon Tversky [2003] nous structurons notre environnement en espaces mentaux, et selon Le Yaouanc et al. [2010] ceux-ci sont triés par ordre de proximité : d'abord l'espace du corps, ensuite celui de l'expérience directe, ensuite l'« espace distant » et enfin l'« horizon ». Les rédacteurs ont tendance à suivre cette hiérarchie ;

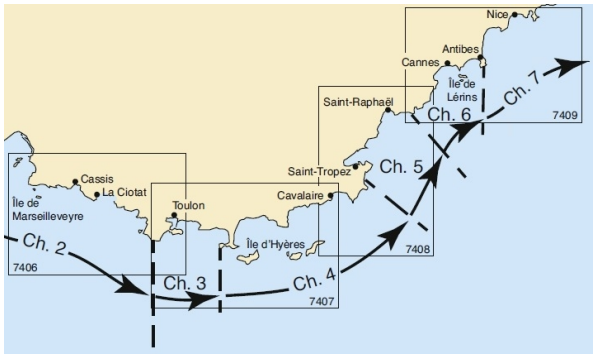


FIG. 2.6: Du Cap Croisette à la frontière italienne : itinéraire structurant du volume D2.1 des *Instructions nautiques*.

- les directions cardinales : selon Nachson et Hatta [2001] et Fuhrman et Boroditsky [2010], le fait que les rédacteurs vivent dans une culture dont la principale direction d'écriture est de gauche-à-droite, ils suivent cette même direction pour décrire les objets. D'ailleurs la structure globale des volumes des *Instructions nautiques* suit un itinéraire qui longe une côte (différente pour chaque volume) dans le sens trigonométrique, ce qui, pour le volume D2.1 dont provient notre corpus d'extraits, correspond aussi à la direction de gauche à droite (cf. fig. 2.6).

2.3 Perspectives

Nous avons comme perspective de faire de l'inférence grammaticale de langages formels étendus, dans deux cas :

Premièrement, un corpus de traces log d'un système d'information bancaire, dans le but de détecter des anomalies. La principale caractéristique de ce corpus est le fait que divers processus écrivent différents types d'information dans les logs, qui peuvent être des fragments de langage naturel (existant sous forme de messages d'erreur dans le code des processus) ou des traces d'erreurs Java (avec toute la hiérarchie des classes) ou des chaînes de caractères faisant partie d'un code. Le corpus est donc l'union de productions de différentes grammaires formelles. Pour permettre l'inférence grammaticale et la fouille de ces données nous allons injecter dans les arbres syntaxiques des relations d'hypéronymie (par exemple, une instance de date aura comme hypéronyme un type « date »). En allant de la racine vers les feuilles on aura donc une augmentation de précision des données. Le langage formel ainsi défini sera muni de la profondeur syntaxique de chaque mot et nous nous servirons de cet indice pour optimiser l'algorithme de prédiction d'anomalie.

Ensuite, un corpus de transcriptions de rapports des équipes de surveillance de certains dispositifs industriels, dans le but de détecter automatiquement des anomalies. Il s'agit d'un langage contrôlé, sur lequel on fera de l'inférence grammaticale tenant compte d'un autre flux de données : des séries temporelles provenant de capteurs intégrés dans les dispositifs. On cherchera donc à combiner des motifs fréquents dans les arbres syntaxiques horodatés avec des motifs extraits des séries temporelles. Nous encadrerons un post-doc de 18 mois sur ce projet.

Enfin, une troisième perspective qui relève du langage visuel : le projet *ELIA Frames* [Chepaitis et al., 2004] a comme ambition de créer une alternative au Braille qui puisse être « lisible » aussi bien par les déficients visuels (en tactile) que par les non-déficients (en visuel) : e n v o i c i u n e x e m p l e (= « EN VOICI UN EXEMPLE »). Parmi ses caractéristiques innovantes il y a l'adaptabilité à toute écriture, et d'ailleurs il y a déjà eu une adaptation à l'écriture arabe (dans le cas du persan) par Siavash [2017]. Notre intention est de procéder à une décomposition simultanée du système ELIA et de l'écriture latine (ainsi que d'autres écritures) en unités élémentaires, de décrire le système ELIA en tant que langage visuel basé sur ces unités, et de l'optimiser sur la base des trois contraintes suivantes : (1) distinctivité tactile maximale, (2) distinctivité visuelle maximale, (3) similarité maximale de la composante visuelle à l'écriture latine (ou autre) traditionnelle. Ce projet vise une amélioration de la vie des handicapés et pourrait profiter de subventions dans ce sens.

Chapitre 3

Similarité, sérendipité, interstices syntaxiques

Ce chapitre se situe dans le domaine du traitement automatique de la langue. Dans la première section nous parlerons de plongements sémantiques obtenus en utilisant Wikipédia en tant que ressource linguistique. Dans la deuxième section nous décrirons le contexte de l'exploration du Web à travers un moteur de recherche et l'utilisation de la notion de saillance pour définir et une similarité spécifique à la situation, et une exploration sérendipiteuse du Web. Enfin, dans la dernière section nous introduirons un procédé d'étude de l'interaction entre pauses/interjections et la structure syntaxique d'une phrase dans un contexte de détection automatique de comorbidités psychiatriques.

3.1 Plongements de mots dans un espace wikipédique catégorisé

Bien avant les plongements de mots utilisés par le *deep learning*, il y a eu des plongements sémantiques de mots, autrement dit des espaces vectoriels de grande dimension dont certains points correspondent à des « mots », et dont la distance euclidienne représente une sorte de similarité sémantique. Ces plongements différaient en plusieurs points :

1. la dimensionnalité de l'espace vectoriel ;
2. ce qu'on entend par « mot » : des formes lexicales dessufixées, des lemmes, des termes (simples ou complexes), des racines dans le cas de l'arabe, etc. ;
3. les propriétés de la distance euclidienne ;
4. la couverture en mots, question étroitement liée à la suivante :
5. le processus de création et de mise à jour de l'espace vectoriel à partir de ressources linguistiques.

3.1.1 De Smart (1964) à l'ESA (2007)

À l'origine de ces méthodes se trouve le VSM (*vector space model*) proposé par Salton [1964] dans le système *Smart*. Salton laisse une certaine flexibilité à la définition de « mot » [Jones, 1981, p. 319], qu'il appelle « terme » : cela peut être une forme lexicale, un syntagme nominal, une classe de thésaurus (= un ensemble de synonymes), une entrée dans une hiérarchie de termes (ce qu'on appellera un *concept* dans une ontologie dont les relations sont hiérarchiques). Pour Salton, les termes constituent une base orthonormale de l'espace vectoriel. Un document est un vecteur dans cet espace,

les coefficients des vecteurs étant des poids représentant l'« importance » du terme vis-à-vis du document. Une requête est également un vecteur dans le même espace, et pour identifier les documents qui servent de réponse à une requête, Salton procède à une comparaison du vecteur de la requête et du vecteur de chaque document (en prenant, par exemple, leur produit scalaire). La particularité de l'approche de Salton est que les termes sont orthogonaux et donc le système ne possède aucune information sur une éventuelle relation entre eux, que ce soit une relation extérieure au corpus (basée sur une ressource linguistique externe), ou inhérente au corpus (et donc calculable à partir des seules données du corpus).

Le VSM peut être formalisé comme suit : soient t_* les termes du corpus, un vecteur de document d sera défini par

$$d := \sum_{i=1}^{N} w(t_i, d) t_i$$

(où N est le nombre total de termes et $w(t_i, d)$ le « poids du terme t_i dans le document d ») et la similarité entre deux documents d, d' est définie par

$$\sigma_{\mathrm{VSM}}(d, d') := \langle d, d' \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} w(t_i, d) w(t_j, d') \langle t_i, t_j \rangle.$$

Mais comme les t_* constituent une base *orthonormale* de l'espace vectoriel, cela donne :

$$\sigma_{\mathrm{VSM}}(d, d') := \sum_{i=1}^{N} w(t_i, d) w(t_i, d').$$

En notation matricielle, si d et d' sont des matrices $N \times 1$ cela revient à écrire

$$\sigma_{\mathrm{VSM}}(d, d') := {}^t d \cdot d'.$$

En 1985, Wong et al. [1985] proposent d'élargir le VSM de Salton en tenant compte de la co-occurrence des termes. Ils appellent leur modèle GVSM (*generalized vector space model*). L'information sur la co-occurrence des termes dans les mêmes documents est stockée dans une matrice G de dimension $N \times N$ et l'équation devient

$$\sigma_{\mathrm{GVSM}}(d, d') := {}^t d \cdot G \cdot d'.$$

Soit $D = (d_*)$ la matrice $N \times M$ dont les documents sont les colonnes. Dans des corpus de grande taille, le nombre de termes N peut être important et les matrices de document sont fatalement éparses. En 1990, Deerwester et al. [1990] introduisent une méthode de réduction de la dimensionnalité basée sur la décomposition en valeurs singulières de la matrice D. Cette méthode est appelée LSA (*latent semantic analysis*), l'aspect « latent » étant la difficulté d'interprétation des dimensions de l'espace vectoriel obtenu. Elle procède comme suit : la matrice D est d'abord décomposée en

$$D = {}^t U_m \cdot S_m \cdot U'_m,$$

où les matrices U_m et U'_m sont unitaires (c'est-à-dire ${}^t U_m \cdot U_m = \mathrm{Id}$ et ${}^t U'_m \cdot U'_m = \mathrm{Id}$) et S_m est diagonale de rang $m = \min(N, M)$. Par une permutation de la base de l'espace vectoriel, on peut

s'arranger pour que les valeurs de la diagonale de S_m soient décroissantes. Alors la méthode LSA prend les k premières valeurs de la diagonale de S_m (avec $l \ll m$) et définit

$$\hat{D} = {}^tU_k \cdot S_k \cdot U'_k.$$

où S_k est la réduction de S_m aux k premières valeurs de la diagonale, et U_k, U'_k les réductions correspondantes de U_m et U'_m. Alors $\hat{D}$ est le modèle de rang k qui est le plus proche de D pour la mesure des moindres carrés.

La similarité entre documents est alors définie par

$$\sigma_{\mathrm{LSA}}(d, d') := d \cdot {}^tU'_k \cdot S_k^2 \cdot U'_K \cdot d',$$

où les vecteurs d et d' sont maintenant représentés dans l'espace réduit de dimension k. D'après Deerwester et al. [1990, p. 13], la méthode LSA se concentre sur la « véritable structure des données », en éliminant le bruit des « détails sans importance ». Selon Gabrilovich et Markovitch [2007, § 4],

1. la méthode LSA ne se base sur aucune « connaissance organisée par l'humain », mais fait son apprentissage à travers la décomposition en valeurs singulières de la matrice des occurrences des mots dans les documents ;
2. elle constitue un plongement des mots dans un espace vectoriel, certes de dimension réduite mais impossible à interpréter sémantiquement, dont ils appelent les dimensions des *concepts latents*.

Pour pallier ces problèmes, Gabrilovich et Markovitch [2007] introduisent en 2007 une analyse sémantique non pas latente mais *explicite* : la méthode ESA (*explicit semantic analysis*). L'idée centrale de cette méthode est le fait que, plutôt que de chercher des concepts artificiels et d'en faire les dimensions de l'espace vectoriel, on pourrait se servir d'une ressource encyclopédique où les concepts sont déjà définis et décrits par des mots. Cela résout les deux problèmes cités, puisque (1) l'organisation des concepts est faite par l'humain, et (2) les dimensions de cet espace ont une interprétation explicite, puisqu'ils proviennent de descriptions de concepts. La ressource choisie par Gabrilovich et Markovitch [2007] est le Wikipédia anglais (400M de mots, 1M d'articles à l'époque de la rédaction de leur article).

ESA fonctionne comme suit : les pages Wikipédia ayant certaines caractéristiques (taille minimum, nombre suffisant de liens entrant et sortant, etc.) sont nettoyées, les mots désuffixés, comptés (une attention particulière est portée à certaines partie de la page, comme le titre, le premier paragraphe, etc.) et pondérés par la mesure tfidf. On obtient ainsi un sac de mots pondérés pour chaque concept (correspondant à une page Wikipédia) et, par transposition, un sac de concepts pondérés pour chaque mot. En prenant les concepts comme dimensions de l'espace vectoriel, chaque mot devient vecteur dans l'espace des concepts, et un document est doté d'un vecteur qui est la somme des vecteurs des mots qu'il comporte.

L'avantage de l'approche ESA vis-à-vis de LSA ou des word2vec actuels est la *transparence* : à chaque dimension de l'espace vectoriel correspond une page Wikipédia et les coefficients de tout vecteur de mot dans cet espace vectoriel peuvent être expliqués par sa mesure tfidf vis-à-vis de chaque page Wikipédia.

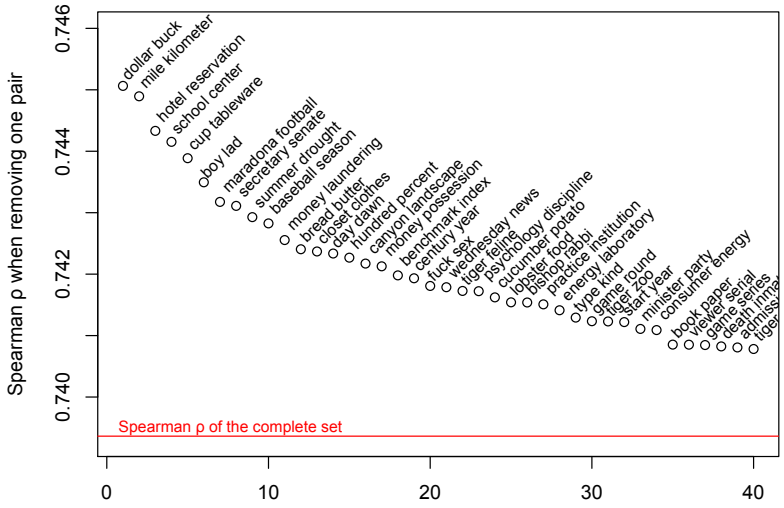
Gabrilovich et Markovitch [2007] évaluent la méthode ESA sur un ensemble de 353 paires de mots (appelé *WordSim353* [Finkelstein et al., 2002 ; Agirre et al., 2009]) dont la similarité sémantique a préalablement été évaluée par seize juges humains. Cet ensemble d'évaluation présente nombreux problèmes : (a) il est spécifique à l'anglais, (b) il n'est pas désambiguïsé (des mots polysémiques

comme *stock* ou *bank* y apparaissent), (c) il comporte des entités nommées dont la comparaison correspond à la vision politique nationaliste israëlienne : $\sigma(Jerusalem, Israel)$ est strictement plus grand que $\sigma(Jerusalem, Palestinian)$, $\sigma(Arafat, terror)$ est strictement supérieur à $\sigma(Arafat, peace)$, etc. Néanmoins il a été utilisé pour l'évaluation de plusieurs travaux sur les mesures de similarité sémantique. Les paires de mots les plus éloignées sémantiquement sont (*king*, *cabbage*) et (*professor*, *cucumber*) ; les paires les plus proches sont (*tiger*, *tiger*) et (*fuck*, *sex*). D'après Gabrilovich et Markovitch [2007, § 3.2], les performances de l'ESA surpassent celles de la LSA : la première atteint un coefficient de Spearman de 0,75 contre 0,56 pour la deuxième.

3.1.2 Optimisation de l'ESA

Dans [YH et Kluyev, 2011] nous avons exploré les limites de l'ESA et nous avons proposé une agrégation de mesures de natures différentes qui fournit de meilleurs résultats que l'ESA seule, sur le même ensemble d'évaluation WordSim353. Après une actualisation des paramètres l'algorithme pour l'adapter aux nouvelles données (entre 2005 et 2011, le Wikipédia anglais est passé de 867k à 4,2M de pages), nous avons obtenu un coefficient ϱ de Spearman de 0,739,

Pour cerner les paires de mots problématiques nous avons calculé ϱ pour les ensembles WordSim353 dont on a supprimé une paire de mots et identifié celles dont la suppression augmente le plus le coefficient de Spearman :



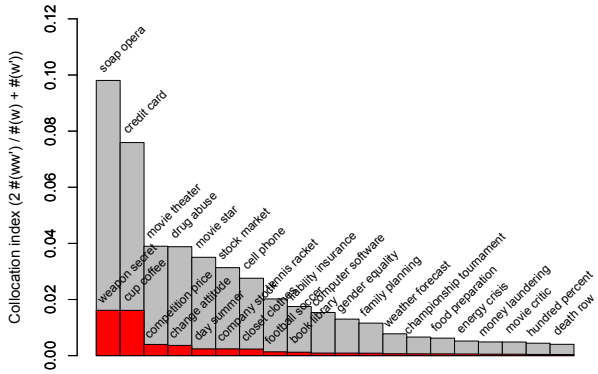
De manière surprenante la plupart des paires « problématiques » de cette liste ont une relation hiérarchique (*dollar/buck* et *boy/lad* sont des synonymes de registre différent, *mile* et *kilometer* sont des unités de mesure, etc.) ou alors sont des collocations (*hotel reservation*, *baseball season*, *money laundering*, etc.).

En ce qui concerne le premier cas, celui des mots en relation hiérarchique de synonymie avec différence de registre, la défaillance de la méthode ESA est due au fait que Wikipédia, n'étant pas un dictionnaire comme Wiktionary, ne comporte pas les mots de registre familier ou d'argot. Cependant ces synonymes se trouvent dans les synsets de WordNet, ainsi nous avons proposé une deuxième mesure basée sur l'inverse de la distance des sommets sous WordNet, et en l'agrégeant avec ESA nous avons obtenu un coefficient de Spearman $\varrho = 0{,}778$, qui dépasse déjà celui de la mesure ESA standard.

En ce qui concerne la collocation, nous nous sommes servis de la ressource GoogleBooks (53 milliards de mots, en ne prenant que les ouvrages publiés après 1970) en définissant les indices de collocation et de collocation inverse :

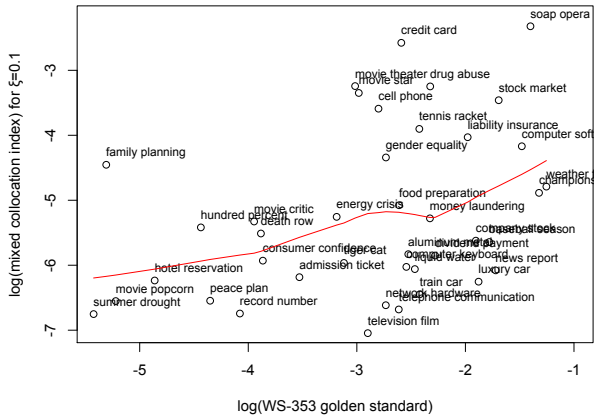
$$\mathrm{coll}(w_1, w_2) := \frac{2\#(w_1 w_2)}{\#(w_1) + \#(w_2)}$$
$$\mathrm{invcoll}(w_1, w_2) := \frac{2\#(w_2 w_1)}{\#(w_1) + \#(w_2)},$$

où $\#(w_1 w_2)$ (resp. $\#(w_2 w_1)$ est le nombre de bigrammes $w_1 w_2$ (resp. $w_2 w_1$) dans GoogleBooks. En calculant les valeurs de ces deux indices pour toutes les paires de WordSim353 nous avons trouvé :



où nous représentons en gris l'indice de collocation (i.e. *soap opera*, *credit card*, etc.) et en rouge l'indice de collocation inverse (*weapon secret*, *cup coffee*, etc.). Plusieurs parmi ces paires faisant partie de celles dont la suppression augmente le coefficient de Spearman, nous avons défini une fonction linéaire des deux indices qui, agrégée à la mesure ESA et à la mesure basée sur WordNet, fournit une performance accrue : la mesure de similarité sémantique agrégée atteint un coefficient de Spearman de 0,787 sur l'ensemble-test WordSim353.

Dans le diagramme ci-dessous nous représentons la relation (logarithmique) entre indice de collocation mixte (direct et inverse) et coefficient de Spearman :



Une partie des bigrammes correspond à la notion de *terme complexe* (cf. L'Homme [2004, chap. 2] mais aussi [YH et Lavagnino, 2011]), d'ailleurs on voit que les paires qui ont le plus fort potentiel d'amélioration de ϱ et le plus important indice de collocation (celles de la partie supérieure

droite de la figure ci-dessus) sont clairement des termes complexes : *soap opera*, *credit card*, *movie theater*, *stock market*, etc., alors que ceux de la partie inférieure gauche le sont peut-être un peu moins clairement : *movie popcorn*, *hotel reservation*, etc. Tout cela nous conduit à penser qu'une extraction terminologique [Daille et al., 1994] éventuellement suivie d'un calcul de *termitude* [Frantzi et al., 1998], pourrait améliorer les performances de l'ESA, en incorporant les pages Wikipédia correspondant aux termes complexes.

3.1.3 Arborification de Wikipédia

L'approche standard de la méthode ESA (que ce soit celle de Gabrilovich et Markovitch [2007], ou la nôtre dans [YH et Kluyev, 2011]) souffre d'au moins trois problèmes :

1. l'évaluation par comparaison avec l'ensemble de 353 paires WordSim353 semble arbitraire,
2. le calcul du tfidf accorde beaucoup d'importance aux mots qui n'apparaissent qu'une ou deux fois dans les pages, mots qui ne participent peut-être que très peu à la définition du « concept » de la page, et
3. la structure hiérarchique de Wikipédia n'est pas exploitée.

Nous avons voulu pallier ces problèmes dans [YH et Kluyev, 2013].

Pour évaluer l'universalité et la pertinence de la méthode ESA, nous avons choisi de traiter le corpus Wikipédia français, dont la taille en 2013 était comparable à celui du Wikipédia anglais de 2005, utilisé pour l'ESA originale. Pour cela nous avons précédé l'étape de désuffixation de Gabrilovich et Markovitch [2007] par une étape de *lemmatisation*, qui était nécessaire, vu la nature flexionnelle du français.

Ensuite nous nous sommes intéressés à la structure hiérarchique de Wikipédia : en effet, les pages Wikipédia sont de deux types : *page d'article* et *page de catégorie*. Il existe des relations d'appartenance de page d'article ou de catégorie à une page de catégorie. Cette relation établit une structure de graphe : on retrouve dans le Wikipédia français (de 2011) 2,78M de pages au total (les sommets du graphe) dont 293k pages de catégorie, 681k arêtes de relation d'appartenance entre catégories et 12,94M arêtes d'appartenance de page d'article à page de catégorie.

Le graphe de Wikipédia est orienté mais non orienté acyclique. Pour détecter les cycles nous avons effectué un grand nombre de marches aléatoires : nous avons trouvé 12 cycles de longueur 3 et 38 cycles de longueur 2 (comme par exemple les catégories ANIMAL et ZOOLOGIE qui s'appartiennent mutuellement). En éliminant 50 arêtes nous avons rendu le graphe orienté acyclique.

Pour introduire l'information fournie par les choix de catégories dans le calcul de l'ESA, nous avons d'abord défini un nouveau type de mésure tfidf, le *tfidf catégoriel*, ensuite nous avons redéfini les vecteurs de page, défini les vecteurs de catégorie comme étant la somme des vecteurs des pages qui sont des descendants d'une catégorie donnée, et enfin une mesure de similarité de relation d'appartenance à l'aide du produit scalaire des vecteurs impliqués. Voici la formalisation de notre approche :

Le tfidf classique $t_p(w)$ du mot (lemmatisé et désuffixé) w par rapport à la page p est défini comme suit :

$$t_p(w) := (1 + \log(f_p(w))) \cdot \log\left(\frac{\#\mathcal{W}}{\sum_{\substack{p\in\mathcal{W}\\ w\in p}} 1}\right),$$

où $f_p(w)$ est la fréquence de w dans la page p, $\mathcal{W}$ est le corpus Wikipédia tout entier en tant qu'ensemble de pages et $\#$ dénote le cardinal. Le deuxième terme (connu comme « idf ») représente le

ratio entre le nombre total de pages Wikipédia et celles qui comportent le terme, c'est ce qui permet de baisser le tfidf de mots très fréquents. On notera que la somme $\sum_{\substack{p\in\mathcal{W}\\ w\in p}} 1$ comporte aussi la page p pour laquelle nous faisons le calcul de $t_p(w)$.

Soit c une catégorie Wikipédia, notons $\mathcal{F}(c)$ la fusion de toutes les pages d'article qui sont des descendants (directs ou éloignés) de c. Selon le niveau hiérarchique de c, $\mathcal{F}(c)$ peut être petit ou alors une partie substantielle de la totalité de Wikipédia. Notre définition de *tfidf catégoriel* modifie le deuxième terme du produit et restreint le dénominateur aux pages qui contiennent *w et qui ne font pas partie de* $\mathcal{F}(c)$:

$$t_c(w) := (1 + \log(\sum_{p\in\mathcal{F}(c)} f_p(w))) \cdot \log\left(\frac{\#\mathcal{W}}{1 + \sum_{\substack{p\in\mathcal{W}\setminus\mathcal{F}(c)\\ w\in p}} 1}\right).$$

Nous pouvons ainsi définir les vecteurs de mot, page et catégorie comme suit :

$$\vec{w} := \sum_{p\in\mathcal{W}} t_p(w)\cdot 1_p \in \mathbb{R}^{\#\mathcal{W}}, \qquad \vec{p} := \frac{\sum_{w\in p} t_p(w)\cdot\vec{w}}{\|\sum_{w\in p} t_p(w)\cdot\vec{w}\|}, \qquad \vec{c} := \frac{\sum_{w\in\mathcal{F}(c)} t_c(w)\cdot\vec{w}}{\|\sum_{w\in\mathcal{F}(c)} t_c(w)\cdot\vec{w}\|},$$

où 1_p est le vecteur unitaire correspondant à la page p.

On peut maintenant définir la *mesure de similarité sémantique des relations d'appartenance* entre une page p et une catégorie c ou entre une catégorie c et une surcatégorie c' en écrivant $\sigma(p,c) := \langle\vec{p},\vec{c}\rangle$ et $\sigma(c,c') := \langle\vec{c},\vec{c'}\rangle$.

Le but de ces opérations est de faire intervenir dans le calcul d'un ESA amélioré les ancêtres des pages auxquelles appartient un mot donné. L'idée est la suivante : un mot pertinent pour une page donnée a des fortes chances d'être présent aussi dans d'autres pages de la même catégorie — inversement, un mot qui apparaît accidentellement dans une page ne se trouvera pas dans les autres pages de la même catégorie. Il faudrait donc augmenter le poids des paires (mot, page) quand le même mot apparaît dans plusieurs pages de la même catégorie. On se propose de définir un *tfidf thématiquement renforcé* qui tient compte de la présence d'un mot w dans une page p et dans les catégories-ancêtres de p.

À ce stade on se heurte à un problème capital : les pages p peuvent avoir plusieurs ancêtres, plus ou moins pertinents. Comment choisir une suite d'ancêtres les plus pertinents possibles ?

Notre hypothèse est que la pertinence du choix des ancêtres dépend de la mesure de similarité sémantique des relations d'appartenance. Plus cette mesure est globalement faible, meilleure sera la pertinence du choix d'ancêtres. Nous avons pondéré les arêtes du graphe de Wikipédia orienté acyclique et nous avons calculé sa forêt orientée couvrante minimale à l'aide de l'algorithme de Chu-Liu et Edmonds publié en 1965 [Gabow et al., 1986]. Comme le graphe du Wikipédia français a un puits global (la page ARTICLE) cette forêt est en réalité un arbre orienté. Notons $\pi^i(p)$ les ancêtres de la page p dans l'arborification ainsi obtenue de Wikipédia. Nous définissons le *tfidf thématiquement renforcé* comme suit :

$$t_{p,\lambda_*}(w) = t_p(w) + \sum_{i\geq 0} \lambda_i t_{\pi^i(p)}(w).$$

où $(\lambda_i)_i$ est une suite décroissante tendant vers zéro et telle que la série $\sum\lambda_i$ converge (dans nos calculs nous avons utilisée $\lambda_i = 2^{-i}$). Ainsi l'effet des ancêtres de p sur la valeur de $t_{p,\lambda_*}(w)$ décroit rapidement quand on s'éloigne de p.

Nous avons défini la mesure TRESA (*thematically reinforced explicit semantic analysis*) de la même manière qu'ESA mais en utilisant le tfidf thématiquement renforcé.

Pour comparer ESA et TRESA nous avons procédé à une évaluation par une classification de textes (avec des SVM comme moteur de classification). Nous avons utilisé un corpus de messages de forums Usenet : 20 forums thématiquement différents, 1 000 messages par forum, un total de 11,4M de mots (67 902 mots distincts). L'ESA standard a donné une précision de 65,58%, alors que la TRESA a donné une précision de 75%, on a donc constaté une amélioration des performances de l'ordre de 14%, ce qui démontre la pertinence de la TRESA vis-à-vis de l'ESA quand la tâche finale est une classification thématique de textes.

Notons que l'arborification de Wikipédia nous a également permis d'identifier la « catégorie la plus pertinente » pour chaque page Wikipédia, ainsi que de trier ses catégories par ordre de pertinence.

3.2 Exploration sérendipiteuse du Web

Les travaux sur la recherche par requête envoyée à un moteur de recherche datent des années soixante du siècle dernier. En 1971, Rocchio [1971] introduit un algorithme basé sur la *pertinence* : le système extrait des termes que l'utilisateur pondère selon leur pertinence, à l'aide de ceux-ci des nouvelles requêtes sont formées. Pour aller plus vite, le système peut également choisir lui-même les termes à utiliser pour l'expansion de requête, on parle alors de *pseudo-pertinence*. Dans [Kluyev et YH, 2012b,a] nous avons appliqué la mesure de similarité agrégée de [YH et Kluyev, 2011] à l'expansion de requête, en prenant les termes les plus proches de ceux de la requête.

Ces travaux étaient plutôt empiriques, et ont fait l'économie d'une modélisation formelle de l'opération de recherche en ligne. Or il s'agit d'une opération complexe à définir : les informations récupérées sont hétèrogènes (URLs, titre, snippets de texte, ...), les résultats obtenus ont une part d'aléatoire (tout en étant relativement stables, du moins on l'espère), ils sont munis d'un rang qui, lui aussi, peut être variable, la relation entre langage de requête et résultats obtenus n'est que vaguement prévisible, etc.

Dans [YH et N'zi, 2019] nous avons proposé une modélisation formelle de la recherche à travers un moteur de recherche en ligne, ce qui nous a permis de définir une notion de *saillance* et un algorithme permettant une exploration sérendipiteuse du Web, à partir de deux termes fixés comme origine et destination de l'exploration.

3.2.1 Modélisation de la recherche en ligne

Une recherche en ligne est une opération qui a comme entrée une requête, appartenant à un *langage de requête*, et comme sortie les « résultats de la requête ». Voici comment nous définissons ces notions :

Soit $\mathfrak{E}$ un ensemble de termes (au sens de Frantzi et al. [1998]) dans une langue donnée. On définit un *langage de requête* $\mathfrak{Q}(\mathfrak{E})$ basé sur $\mathfrak{E}$ (ou $\mathfrak{Q}$ pour simplifier) le langage formel basé sur l'alphabet $\mathfrak{E} \cup \{(,), \wedge, \vee, \neg\}$ et sur les règles

$$\begin{aligned} &S \to W, \\ &W \to W \wedge W \mid W \vee W \mid \neg W \mid (W), \\ &W \to m, \end{aligned}$$

où $m \in W$ et S est l'axiome de départ.

On définit les paramètres suivants : N le nombre de requêtes identiques envoyées aux instants $t_1, \ldots, t_N$, et M le nombre de lignes de résultats dont on va tenir compte.

Nous traitons différents types d'unités d'information : des URL, des termes, des synsets WordNet, des concepts dans une ontologie. Pour chaque type d'unité on définit un *objet d'information* de dimension M comme une paire (c, v), où c est une unité et v un *vecteur de poids*, c'est-à-dire un élément de $\mathbb{R}^M$. On appelle c le *contenu* de l'objet d'information. On note $\mathbb{O}_*$ l'ensemble des objets d'information de type $*$.

Par exemple, imaginons que t est le terme « téléphone portable » et qu'il apparaît dans la 2e et la 5e ligne de résultats obtenus avec des poids 0,1 et 0,6 (les poids restent à définir), alors l'objet d'information est

$$(\text{« téléphone portable »}, (\underbrace{0, 0.1, 0, 0, 0.6, 0, \ldots, 0}_{M \text{ nombres}})).$$

Définissons maintenant une *recherche Web* comme étant l'opération qui à un mot q du langage formel $\mathfrak{Q}$, un type d'objet d'information $*$ et N instants temporels $t_1 < t_2 < \cdots < t_N$ associe $\phi_{*,t_1,\ldots,t_N}(q) \in 2^{\mathbb{O}_*}$, c'est-à-dire un ensemble d'objets d'information, obtenus de la manière suivante : si $I = (c, v)$ est un objet d'information appartenant à $\phi_{*,t_1,\ldots,t_N}(q)$, alors $v = \frac{1}{N}\sum_{j=1}^{N} \lambda_{i,j}(I)$ où $\lambda_{i,j}(I)$ est soit la présence (1 si présent, 0 si absent) de I en rang i et à l'instant j soit une valeur calculée à partir de I, i et j.

En guise d'exemple, prenons $N = 10$, $M = 3$, et imaginons que $\lambda_{i,j}(I)$ soit la présence de l'objet I (de type URL) en rang i et à l'instant t_j. Imaginons ensuite que l'on récupère quatre URLs différentes α, β, γ, δ de la manière suivante :

Temps	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
Rang 1	α	α	α	β	α	β	α	β	α	β
Rang 2	β	β	γ	α	β	α	β	α	β	α
Rang 3	γ	γ	β	γ	γ	γ	δ	γ	γ	γ

Alors le résultat de la recherche $\phi_{U,t_1,\ldots,t_{10}}(q)$ sera l'ensemble des quatre objets d'information suivants :

$$\{(\alpha, (0.6, 0.4, 0)), (\beta, (0.4, 0.5, 0.1)), (\gamma, (0, 0.1, 0.8)), (\delta, (0, 0, 0.1))\}.$$

3.2.2 Stabilité de la recherche Web

On peut se poser la question de l'impact du caractère aléatoire des requêtes Web : peut-on s'affranchir de la donnée des instants temporels $t_1, \ldots, t_N$ quand on parle des résultats d'une requête ?

Pour cela il a fallu définir une *distance* entre ensembles d'objets d'information. Nous faisons cela de manière canonique : soit $I = \{(c_1, v_1), \ldots, (c_k, v_k)\}$ et $I' = \{(c'_1, v'_1), \ldots, (c'_{k'}, v_{k'})\}$. Nous rénumérotons les (c'_i, v'_i) de manière à ce que si $c_i = c'_j$ alors c'_j devienne c'_i et, inversement, pour un même j, si c_j et c'_j existent alors $c_j = c'_j$. Cela nous permet de comparer les poids des éléments communs. Il ne reste qu'un détail à régler : pour chaque i, v_i est un vecteur de $\mathbb{R}^M$, on peut comparer les vecteurs de poids en accordant une importance égale à tous les rangs ou en accordant une importance moindre aux rangs élevés. En général, si $f : \mathbb{R}^M \to \mathbb{R}_+$ est une fonction telle que $f(v) = 0 \Rightarrow v = (0, \ldots, 0)$, on peut définir la distance de Soergel [Willett et al., 1998]

$$d_f(I, I') := \frac{\sum_{i=1}^{K} |f(v_i) - f(v'_i)|}{\sum_{i=1}^{K} \max(f(v_i), f(v'_i))}.$$

Prenons les deux fonctions suivantes :

1. d_{ranked} pour $f(v_i) := \sum_{j=1}^{M} \frac{v_{i,j}}{j}$, où $v_{i,j}$ est le j-ème élément du vecteur v_i ;
2. d_{unranked} pour $f(v_i) := 1$ s'il existe au moins un j tel que $v_{i,j} > 0$.

Notons que d_{unranked} est la distance de Jaccard des ensembles $\{c_1, \ldots, c_k\}$ et $\{c'_1, \ldots, c'_{k'}\}$.

Il ne reste plus qu'à définir la stabilité d'une recherche Web :

Soit d une distance sur $\mathbb{O}_*$ telle que $\mathrm{Im} d \subset [0,1]$. Soient des limites temporelles T_0 et $T_E > T_0$. On dira que $\phi_{U,t_1,\ldots,t_N}$ est *d-stable* sur $[T_0, T_E]$ si

$$\max_{\substack{T_0 \le t_1 < \cdots < t_N \le T_E \\ T_0 \le t'_1 < \cdots < t'_N \le T_E}} d(\phi_{U,t_1,\ldots,t_N}(q), \phi_{U,t'_1,\ldots,t'_N}(q)) \le 0{,}05,$$

pour tout $q \in \mathfrak{Q}$.

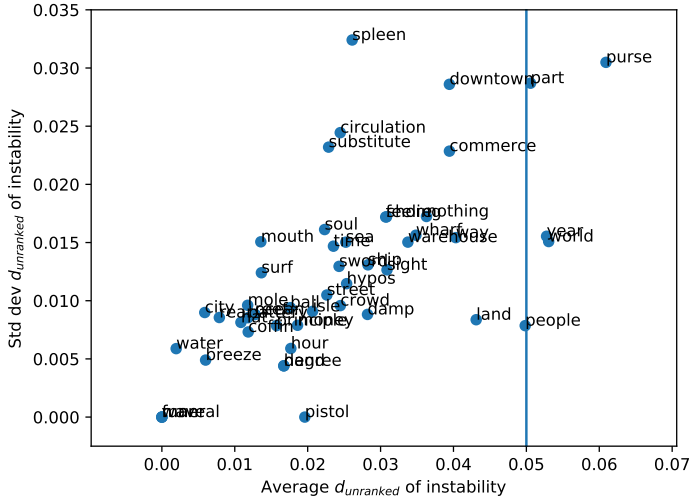
Vérification expérimentale

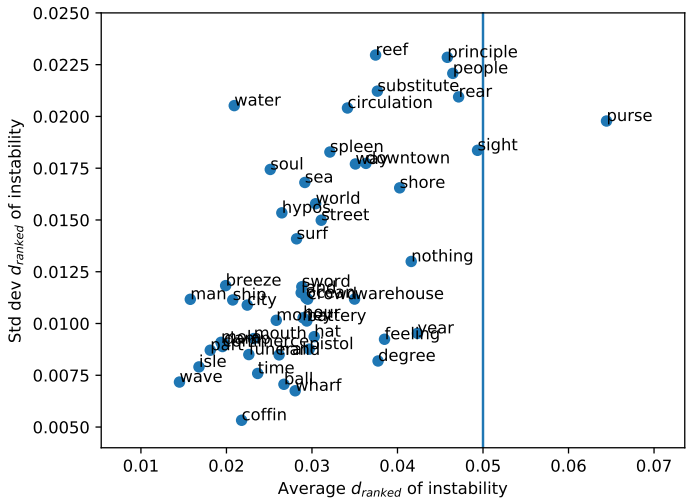
Nous avons vérifié expérimentalement que l'opération ϕ_U basée sur le moteur de recherche Bing est une recherche Web aussi bien d_{ranked}-stable que d_{unranked}-stable.

Pour cela nous avons posé $N = 10$ et $M = 50$ et nous avons pris l'ensemble de mots suivant :

year, money, purse, nothing, shore, part, world, way, spleen, circulation, mouth, damp, soul, coffin, warehouse, rear, funeral, hypos, hand, principle, street, people, hat, time, sea, substitute, pistol, ball, sword, ship, man, degree, feeling, ocean, city, wharf, isle, reef, commerce, surf, downtown, battery, mole, wave, breeze, hour, sight, land, crowd, water,

qui sont les 50 premiers substantifs du roman *Moby Dick*. Nous avons lancé deux séries de 10 recherches Web pour chaque mot, donc au total 200 requêtes par mot, le tout dans un intervalle de 12 heures. Nous avons calculé les deux distances entre les mêmes mots dans les deux séries. Voici les résultats :





On peut donc légitimément écrire ϕ_U sans mentionner les t_i.

3.2.3 Saillance et treillis de sérendipité

Nous pouvons maintenant parler de ϕ_* comme d'une application de $\mathfrak{Q}$ dans $2^{\mathbb{O}_*}$. Dans la suite nous allons nous concentrer sur les objets d'information de type terme ($\mathbb{O}_T$) et sur les requêtes du type $\bigwedge t^{(i)}$ (conjonctions de termes).

On dira que la recherche $\phi_T(\bigwedge_{i=1}^{n} t^{(i)})$ est *valide* si elle contient des objets d'information $(t_1, v_1), \ldots, (t_n, v_n)$ tels que pour chaque $j \in \{1, \ldots, M\}$, soit tous les $v_{i,j}$ sont strictement positifs, soit ils sont tous nuls. Cela signifie que dans tous les résultats on retrouve tous les termes de la requête.

Toute requête qui contient au moins un résultat avec tous les termes requis peut devenir valide si on enlève tous les résultats qui ne contiennent que quelques-uns des termes.

On arrive maintenant à la définition de la notion de *saillance* :

Soit q une requête et $\phi_T(q)$ une recherche valide. Soit t un terme. Alors on dira que t est *q-saillant* s'il existe un vecteur de poids non nul v tel que $(t, v) \in \phi_T(q)$.

Autrement dit, un terme t est *saillant pour une requête, s'il apparaît dans ses résultats*. Par définition cela est vrai pour les termes de la requête elle-même.

Nous considérons la saillance comme une *relation de similarité sémantique* entre les termes de q et t.

Soit deux termes t et t'. Nous allons appliquer la notion de saillance pour construire un treillis de termes avec t comme plus petit élément et t' comme plus grand élément :

On appelle *treillis de sérendipité* entre t et t' un treillis ayant les propriétés suivantes :

1. tous les chemins $t = t_1 \succ \cdots \succ t_m \succ t'$ sont de même longueur m (la hauteur du treillis) ;
2. tout élément t_i d'un tel chemin (avec $t_i \neq t'$) n'est pas $(t_1 \wedge \cdots \wedge t_{i-1})$-saillant ;
3. t' est $(t_1 \wedge \cdots \wedge t_m)$-saillant.

L'idée est que les chemins ainsi construits permettent d'aller de t à t' en passant uniquement par des termes non-saillants, et donc « surprenants », tout en conservant à la fin le fait que t' est saillant pour tous les termes précédents du chemin. Sur la fig. 3.1 nous donnons un exemple de treillis de sérendipité pour la paire de mots *man* / *circulation*.

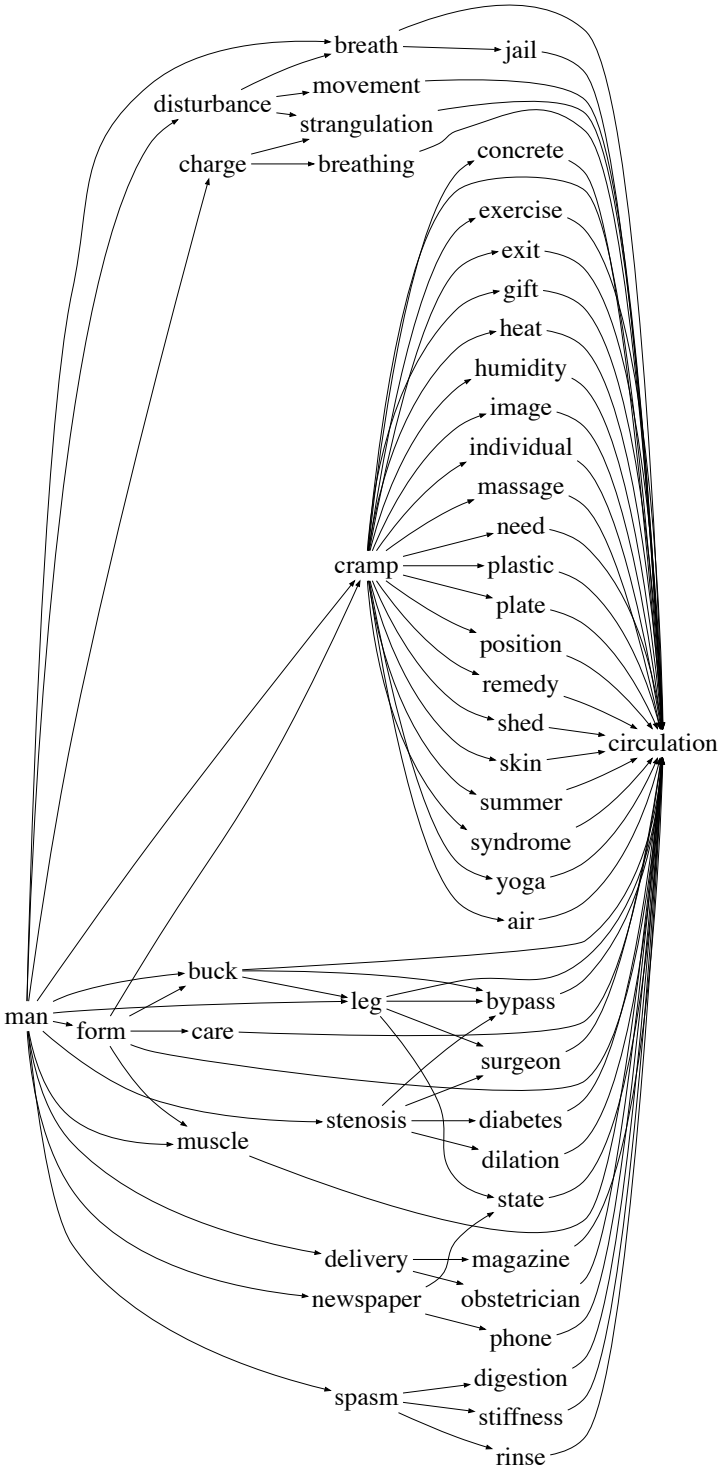


FIG. 3.1: Treillis de sérendipité pour la paire de mots *man* / *circulation*.

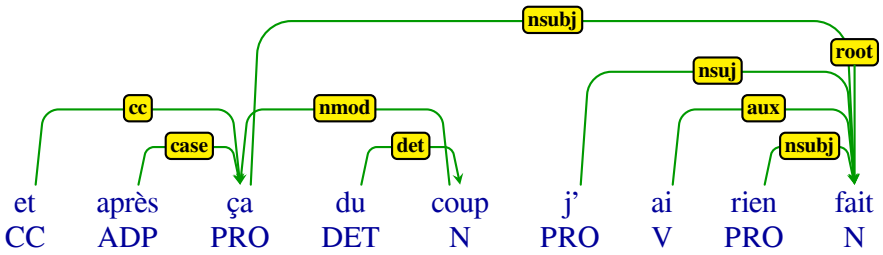
3.3 Interstices syntaxiques et détection automatique de comorbidités psychiatriques

Les travaux décrits dans cette section sont les prémisses d'un projet en cours sur la détection précoce de la schizophrénie. Le corpus utilisé est un ensemble d'entretiens de bilan psychiatrique entre un soignant et un soigné. Comme au moment de la rédaction de [YH et al., 2020] il était trop tôt pour savoir si les personnes interviewées ont présenté des symptômes de schizophrénie, nous nous sommes concentrés sur la recherche d'indicateurs qui corroborent le diagnostic comoborbiditaire fait par le soignant.

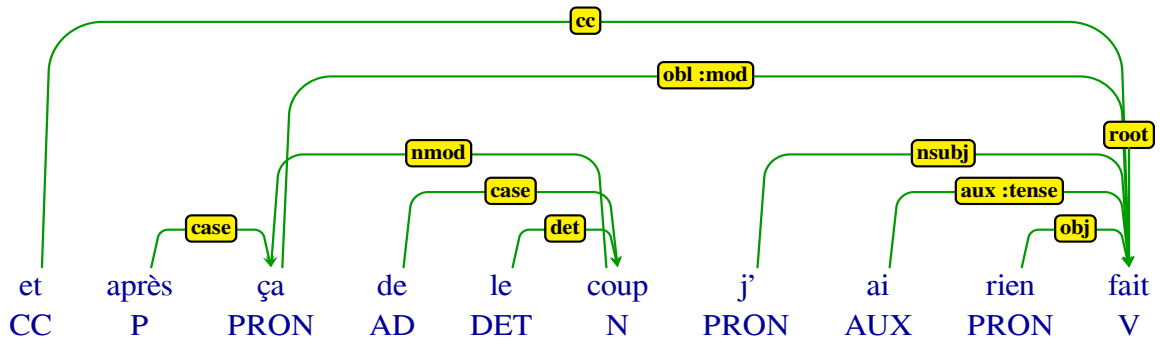
Dans les travaux déjà existants du domaine, les indicateurs linguistiques détecteurs de pathologie psychiatrique relèvent surtout de la phonétique, de la sémantique et de l'analyse du discours. Nous avons étudié un large spectre d'indicateurs mais nos meilleurs résultats font intervenir la syntaxe, et plus particulièrement l'interaction entre interjections ou pauses et structure syntaxique.

3.3.1 Analyse syntaxique de texte informel

Les analyseurs syntaxiques du français partent du principe que les énoncés sont des phrases régulières et que tous les éléments fournis ont un rôle à jouer dans l'arbre syntaxique de la phrase. Ainsi une phrase comme « et après ça du coup j'ai rien fait » (qui provient de notre corpus) a induit quasiment tous les analyseurs syntaxiques en erreur. Voici le résultat de l'analyse syntaxique par l'outil spaCy [Honnibal et Montani, 2017] :

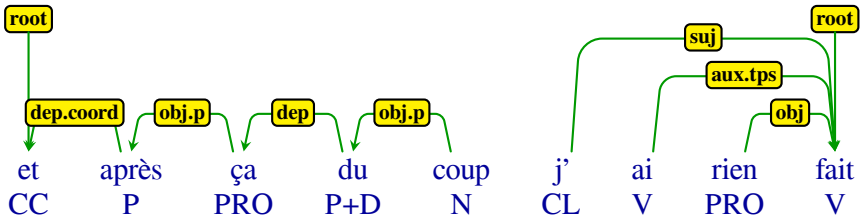


Cet outil a considéré que le participe passé « fait » était un nom et qu'il était la racine de la phrase (dans le sens de l'arbre des dépendances), le verbe étant « ai » et le sujet « ça ». Stanza [Qi et al., 2020] (entraîné sur le corpus GSD) réussit beaucoup mieux à capter la syntaxe de cet énoncé informel :



puisqu'il reconnaît bien « fait » comme étant le verbe, et donc une racine potentielle de l'arbre, mais il essaie à tout prix de relier tous les mots de la phrase et il fait de « ça » un modificateur gouverné par la racine, alors qu'en réalité il est plutôt gouverné par la préposition « après ».

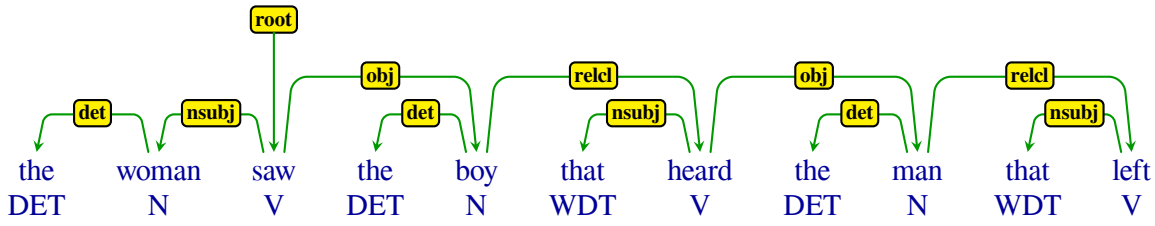
La solution nous a été donnée par l'outil grew [Guillaume et Perrier, 2015], basé sur la réécriture des arbres syntaxiques [Bonfante et al., 2018] :



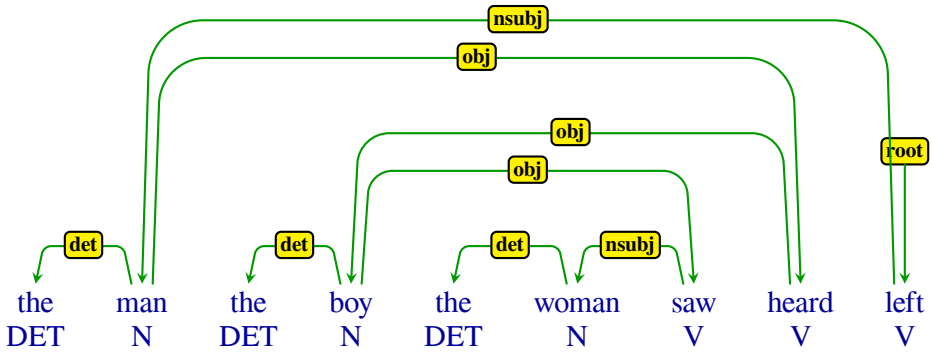
Cet outil sépare les deux phrases et n'hésite pas à considérer que l'énoncé a deux racines. Cette approche bottom-up de l'analyse syntaxique est essentielle pour la mesure d'importance des interstices syntaxiques.

3.3.2 Croisement interstitiel de dépendances syntaxiques

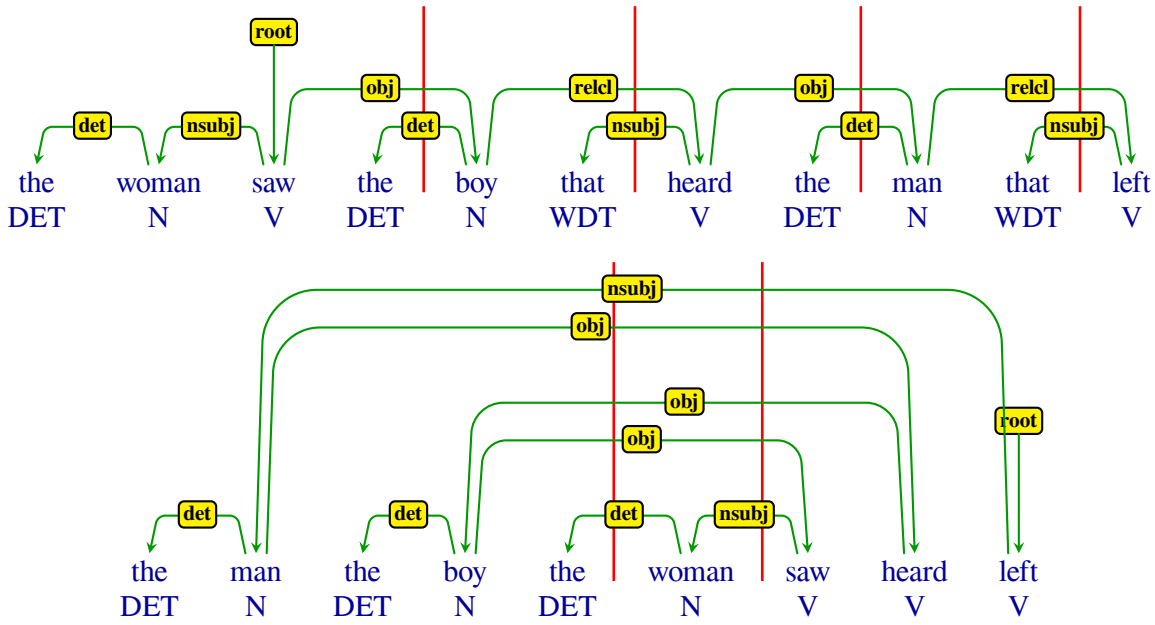
Les arbres syntaxiques ont déjà servi du point de vue des sciences cognitives pour évaluer la complexité linguistique : Liu [2008] définit la *distance moyenne des dépendances* DMD en tant qu'indicateur qui lui permet d'attester et de quantifier le caractère incompréhensible de la phrase *the man the boy the woman saw heard left* qui est pourtant grammaticale. En effet, elle a une valeur de DMD de 3 alors que la phrase sémantiquement équivalente *the woman saw the boy that heard the man that left* a une valeur de DMD bien moindre : 1,4. Comparons les arbres syntaxiques des deux phrases :



et



On appelle *croisement interstitiel* le nombre de dépendances que va intersecter une droite verticale tracée entre deux mots. Dans les diagrammes ci-dessous nous avons tracé les interstices de croisement interstitiel maximal :



Nous voyons que dans le premier cas le croisement interstitiel maximal est de 2 alors que dans le deuxième cas, il est de 5. Nous allons utiliser la présence de pauses ou d'interjections dans un croisement interstitiel comme indicateur de désorganisation linguistique du patient.

3.3.3 Évaluer les pauses et les interjections

Les entretiens enregistrés sont segmentés, transcrits et horodatés. Les deux flux de données (son et texte) sont fournis en entrée à SPPAS [Bigi, 2015] qui produit des listes de phonèmes, de mots en orthographe standard et de mots phonémisés, toutes horodatés. Mais SPPAS ne capte pas les pauses. Nous passons par une version filtrée anti-bruit du texte sous Praat [Boersma et Weenink, 2001] pour détecter les pauses et les introduire dans les données horodatées par SPPAS. Praat nous fournit aussi l'énergie, le pitch, les F1 et F2, que nous alignons avec les phonèmes et les mots.

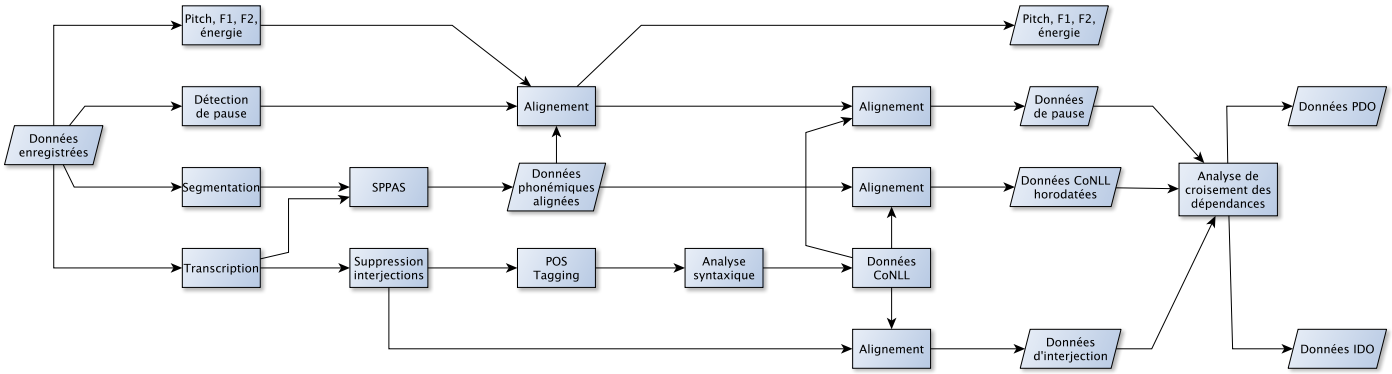


FIG. 3.2: Le processus d'extraction de données.

Dans un flux de données différent (cf. fig. 3.2) nous supprimons les interjections et effectuons une analyse morphosyntaxique avec Talismane [Urieli et Tanguy, 2013] et grew [Guillaume et Per-

rier, 2015] pour obtenir des données CoNLL. Nous nous retrouvons alors avec deux flux de données complémentaires : nous les alignons à l'aide de l'algorithme de Needleman et Wunsch [1970], implementé dans bioPython. Ainsi nous aboutissons à des données CoNLL horodatées avec pauses et interjections.

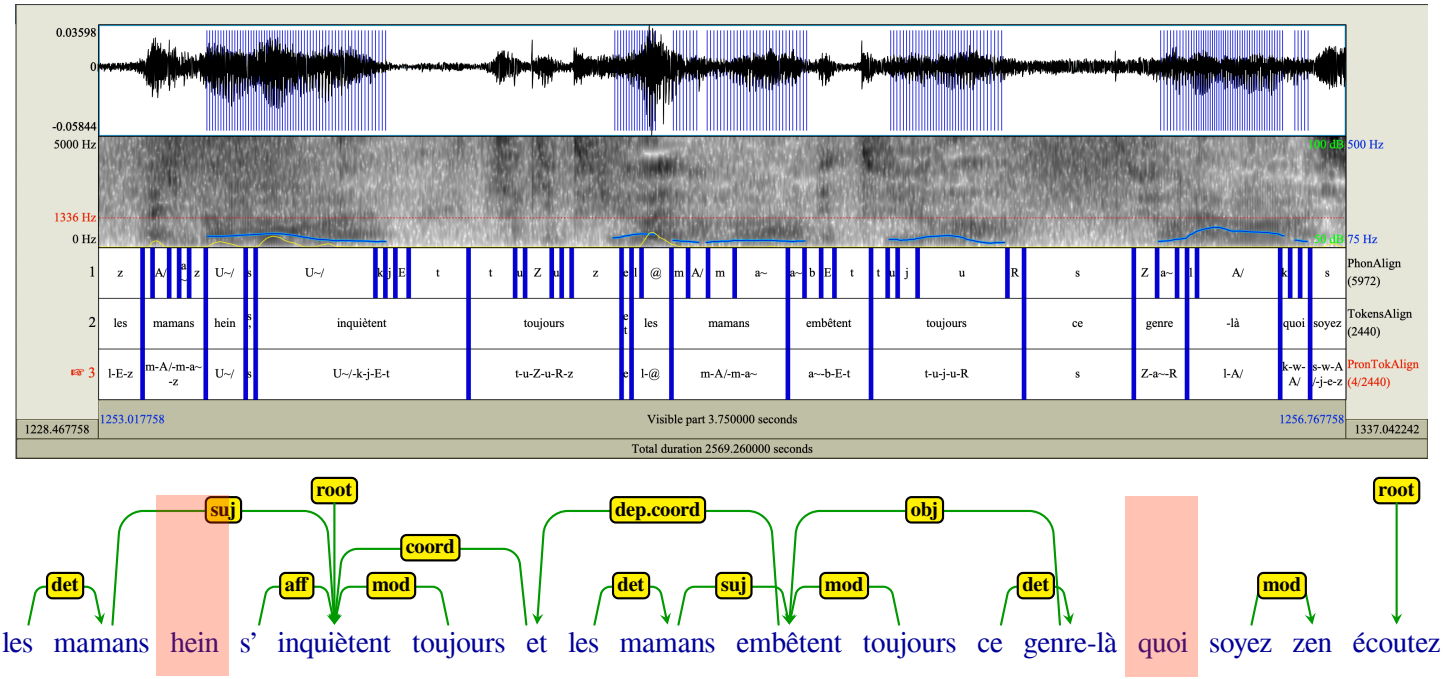


FIG. 3.3: Interface Praat et analyse syntaxique de l'énoncé « les mamans...».

Dans la fig. 3.3 nous présentons la fenêtre de Praat avec les différentes couches d'information ainsi que l'analyse syntaxique par grew avec réinsertion des pauses et interjections de l'énoncé

les mamans s'inquiètent toujours et les mamans embêtent toujours ce genre-là quoi soyez zen écoutez.

L'interjection primaire « hein » intersecte une dépendance de type sujet alors que l'interjection secondaire « quoi » n'intersecte aucune dépendance.

Pour chaque énoncé et pour un ensemble S de types de relations de dépendances nous définissons les indices PIDC_S (*croisement interstitiel des pauses sur l'ensemble S*) et IIDC_S (*croisement interstitiel des interjections sur l'ensemble S*) de la manière suivante :

$$\mathrm{PIDC}_S := (\#\text{croisements de dépendances dans } S) \times \frac{\text{durée des pauses}}{\text{durée de l'énoncé}},$$
$$\mathrm{IIDC}_S := (\#\text{croisements de dépendances dans } S) \times \frac{\text{durée des interjections}}{\text{durée de l'énonce}}.$$

Pour déterminer les relations de dépendance à tester nous nous sommes tournés vers le *French Treebank Corpus* [Abeillé et al., 2003] qui est la plus importante ressource de ce type pour le français. Les relations les plus fréquentes sont :

nature	fréquence	dist. moyenne dép./gouv.
mod	120 741	1,1937
obj.p	90 400	1,7511
det	85 154	1,1987
suj	35 402	4,2315

La dépendance « mod » (modificateur) est très fréquente mais peut prendre plusieurs formes : dans 26% des cas le mot dépendant est un adjectif, dans 22% de cas du FTB il s'agit d'une préposition, dans 20% des cas d'un adverbe. Ces mots peuvent tous se trouver à une certaine distance de leur gouverneur, de ce fait l'existence d'une pause ou d'une interjection entre dépendant et gouverneur n'est pas nécessairement signifiante.

Par contre, la dépendance « obj.p » (objet prépositionnel) est l'équivalent du *case goverment* (pour les langues avec cas) et donc, selon Osborne [2019, p. 142], il s'agit techniquement d'une dépendance plutôt morphologique que syntaxique. Elle est très stable en termes de POS tag (86% des dépendants sont des noms) et la distance entre dépendant et gouverneur est très faible (1,7511 en moyenne). Sa nature morphologique et ses propriétés de stabilité et de proximité au gouverneur nous poussent à émettre l'hypothèse que le croisement d'une telle dépendance et d'une interjection ou d'une pause est susceptible d'indiquer une désorganisation linguistique.

La dépendance « det » (déterminant) est également candidate à révéler une désorganisation : la liste des déterminants est très courte et ils sont très proches de leur gouverneur (1,1987 en moyenne, c'est la distance moyenne la plus courte parmi toutes les relations).

Enfin, la relation « suj » (sujet) est importante puisque toute phrase verbale française (sauf en mode impératif) possède un sujet. Nous l'avons incluse dans le groupe de dépendances à étudier, malgré sa distance moyenne relativement élevée entre dépendant et gouverneur (4,2315 en moyenne).

3.3.4 Résultats

Au moment de la rédaction de [YH et al., 2020] notre corpus était encore assez réduit : il comportait huit entretiens avec des patients, d'une durée qui se situait entre 25 et 63 minutes. Nous avons regroupé les comorbidités constatées par les soignants dans trois groupes :

THY *Désordres thymiques* ;
ANX *Désordres anxieux* ;
ADD *Désordres addictifs*.

Nous avons effectué une correlation de Spearman entre les groupes de comorbidité et les indicateurs PIDC et IIDC, voici les principaux résultats :

Indicateur	Groupe dépendances	Groupe comorbidités	ϱ	p-valeur
PIDC	{obj.p}	**ADD**	**0,8660**	0,0054
PIDC	{det}	**ADD**	0,7735	0,0254
PIDC	toutes	**THY**	0,7042	0,0512
IIDC	{obj.p}	**ADD**	0,8248	0,0117
IIDC	{det}	**ADD**	0,7285	0,04
IIDC	{det, suj}	**ANX**	−0,8247	0,0117
IIDC	{suj}	**ANX**	**–0,8450**	0,0080
IIDC	toutes	**THY**	−0,6730	0,0671

Notre première constatation importante est que pour les dépendances {obj.p} et {det} nous obtenons des résultats très proches aussi bien pour les pauses que pour les interjections, alors que ces deux phénomènes paralinguistiques sont distincts et ont été mesurés de manières différentes (les pauses ont été détectées et mesurées automatiquement par Praat, alors que les interjections ont été saisies lors de la transcription, et horodatés selon le processus de la fig. 3.2).

La dépendance {obj.p} est un indicateur très fort ($\varrho > 0{,}82$ avec un $p = 0{,}012$) pour l'appartenance à **ADD**. La dépendance {det} est également un indicateur d'appartenance à **ADD** mais dans une moindre mesure (un ϱ autour de 0,75, avec une p-valeur entre 0,025 et 0,04).

Autre fait marquant : combiner IIDC avec la dépendance {suj} nous fournit un indicateur très fort contre l'appartenance à **ANX** ($\varrho < -0{,}824$ avec une p-valeur de $= 0{,}012$).

Enfin, en utilisant la totalité des dépendances, le PIDC s'avère être un indicateur (moyen et moyennement fiable) *pour* l'appartenance à **THY** (la p-value dépasse très légèrement 0,05) alors que le IIDC s'avère être un indicateur (tout aussi moyen) *contre* l'appartenance à ce même groupe de comorbidités (avec une p-value de 0,0671).

En ne prenant que les résultats de p-valeur acceptable, nous pouvons énoncer les résultats comme suit : *les membres du groupe **ADD** ont tendance à placer des pauses ou des interjections entre une préposition et le nom gouverné ou entre un déterminant et le nom qui le gouverne, et les membres du groupe **ANX** ont tendance à ne pas placer d'interjections entre un sujet et le verbe qui le gouverne.*

Le premier résultat peut refléter la prévalence des comportements addictifs aux risques psychotiques [Valmaggia et al., 2014]. Comme mentionné ci-dessus, l'occurrence fréquente d'une interjection ou d'une pause entre une préposition et le nom qu'elle gouverne ou entre un déterminant et un nom est un indicateur de désorganisation linguistique, et celle-ci est un des symptômes psychotiques décelés chez les patients à risque [Fusar-Poli et al., 2013]. D'autre part, l'intensité des symptômes psychotiques est corrélée avec l'importance des comportements addictifs [Korver et al., 2010].

Le deuxième résultat peut être expliqué par une tendance des patients anxieux d'éviter des interruptions de flux de parole dans une conversation où ils sont sujets au jugement de leur interlocuteur [Iverach et Rapee, 2014].

3.4 Perspectives

Nous allons poursuivre le projet d'analyse de données textuelles d'entretiens psychiatriques, dans le but de la détection précoce de la schizophrénie. En été 2020 nous avons récolté un corpus d'entretiens de même type que les entretiens psychiatriques mais sur une population de personnes en dehors du système de santé, ce corpus nous servira comme corpus de groupe témoin. Nous avons comme projet de combiner les méthodes d'annotation de Lacheret-Dujour et al. [2019] avec nos indicateurs interstitiaux pour une meilleure prise en compte de la microsyntaxe du texte informel.

D'autre part nous avons démarré un projet sur la pseudonymisation appliquée à un corpus de messages électroniques qui aboutira sur la direction d'une thèse. Les très récentes directives de l'Union européenne sur la protection de la vie privée demandent une anonymisation des données textuelles en possession des entreprises. Celle-ci peut avoir des effets néfastes sur l'utilisabilité des données pour alimenter les algorithmes de traitement automatique de la langue. Pour remédier à ce type de « pollution » des données tout en restant dans les limites de la légalité, la *pseudonymisation* est une méthode d'anonymisation qui produit des données linguistiquement cohérentes [Bourka et al., 2019]. Nous allons traiter ce problème comme un problème d'optimisation de transformation tex-

tuelle sur la base de deux contraintes : l'anonymisation non-réversible et la cohérence linguistique (et, en particulier, sémantique).

Enfin, nous encadrons actuellement une thèse sur le dipôle discursif que constituent les CV et les offres d'emploi. Le but de cette thèse est de modéliser ce dipôle et de développer des méthodes de fouilles de texte adaptées à l'évaluation de la conformité d'un CV vis-à-vis d'une offre d'emploi. Comme il sera montré dans la thèse, les CV et les offres d'emploi sont rédigés dans un *langage de spécialité*, et le facteur de mimétisme et de réutilisation (officieuse) de ressources disponibles sur le Web en fait presque un *langage contrôlé*. Néanmoins l'adaptation d'un CV (qu'il soit rédigé *ab ovo* par le candidat ou récupéré sur le Web) à une offre d'emploi spécifique instaure la relation de *dipôle discursif* dont l'étude sera, nous l'espérons, la principale innovation apportée par cette thèse.

Chapitre 4

Exercices de vulgarisation

> Le tact dans l'audace, c'est de savoir *jusqu'où on peut aller trop loin*.
>
> Jean Cocteau, *Le coq et l'arlequin*

Si, comme disait Halmos [1970],

> une bonne méthode pour préparer un texte mathématique est de s'imaginer expliquer le sujet à un·e ami·e lors d'une longue promenade dans la forêt,

image qui nous fait irrésistiblement penser à Schubert (*Der Wanderer*), à Goethe (« Über allen Gipfeln ist Ruh...») ou à Thoreau (*Walden*), la vulgarisation ajoute une nouvelle dimension à cette transmission de savoir : l'audace. Audace de l'auteur qui vise à générer de la curiosité et de l'ambition chez le lecteur.

Nous avons pu dernièrement rédiger deux articles de vulgarisation pour l'excellente revue *Quadrature*, revue de « mathématiques pures et épicées » comme on peut lire sur son site, qui s'adresse à un public assez vaste, allant des lycéens aux premier et deuxième cycle universitaires, et au-delà...

Mais avant de décrire ces deux articles, quelques mots sur l'activité de vulgarisation.

4.1 La vulgarisation

Pelay et Artigue [2016] affirment que « la didactique des mathématiques en France s'est modérément intéressée jusqu'ici à l'étude des contextes de vulgarisation », en effet ils n'ont relevé en tout et pour tout que 3 thèses et 6 articles de colloque sur le sujet (et aucun article dans la prestigieuse revue *Recherches en Didactique des Mathématiques*). D'autres, comme Rittaud [2015] sont tellement déçus par le manque d'intérêt des didacticiens envers la vulgarisation qu'ils proposent la création d'une nouvelle discipline, la *vulgaristique*. Nous n'allons pas entrer dans ce débat. Dans cette section nous allons brièvement donner quelques éléments qui caractérisent la vulgarisation et dont nous allons nous servir par la suite.

Dans sa thèse de doctorat, Sousa Do Nascimento [1999] fournit une liste d'intentions possibles d'une action de diffusion (dans le cas particulier des mathématiques) :

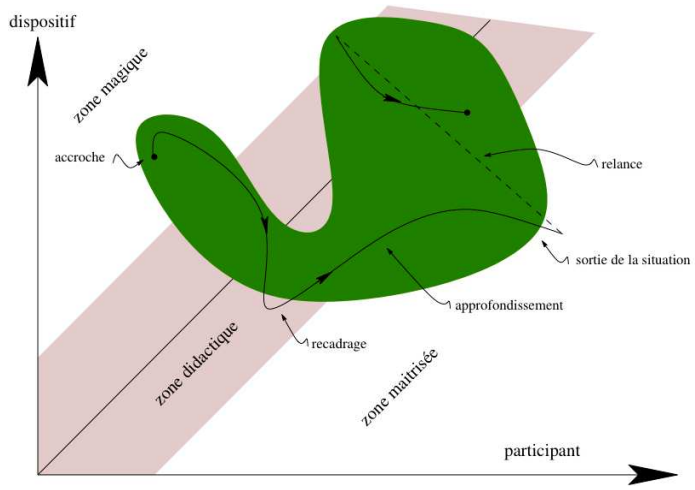


FIG. 4.1: Figure tirée de Pelay et Mercat [2012], Pelay et Boissière [2015] et Pelay et Artigue [2016]. Légende originale : Une activité est vue comme une trajectoire individuelle et collective dans l'espace des connaissances.

Intentions	Enjeux	Rôle de l'animateur
Élucidation, conscientisation, démystification	Valeurs	Militant
Production	Procédures (règles, normes, etc.)	Technicien
Médiation	Culture scientifique et technique partagée	Médiateur
Instruction	Connaissances scientifiques	Instructeur
Loisirs	Plaisir, sensibilisation	Amuseur

Ce tableau montre que le vulgarisateur (de même que l'enseignant) peut, à tour de rôle, être médiateur ou instructeur ou militant (voire même amuseur). Pour tenir compte de cette richesse de méthodes possibles, Pelay [2011] élargit la notion classique de contrat didactique [Brousseau, 1998] en « contrat didactique et ludique ». Dans le cas de la diffusion textuelle, les possibilités d'instaurer des jeux sont fortement réduites, et la part de plaisir relève plutôt de l'émerveillement, de la curiosité, voire du défi.

Rittaud [2015] résume en six points les différences entre enseignement (institutionnalisé) et vulgarisation : (a) la *mise en scène* (cet élément concerne surtout les animations et s'applique un peu moins au texte) ; (b) la *captivité du public* : ce sera un de nos soucis majeurs tout au long de ce chapitre ; (c) la *pérennité du discours*, dans le sens qu'un animateur agit ponctuellement et ne peut donc pas « rectifier le tir » comme le ferait un enseignant dans le cadre d'un cours de longue durée, encore un point qui ne concerne pas l'écrit ; (d) la *mise en perspective*, « cette liberté que la vulgarisation peut davantage se permettre » ; (e) l'absence de *dimension hiérarchique*, dans le sens où un vulgarisateur n'évalue ni ne juge ses auditeurs ; et enfin (f) l'*adaptabilité du contenu* : « le vulgarisateur est plus souvent libre d'orienter son discours ou son atelier vers des éléments qui, sur le moment, apparaissent retenir davantage l'attention, ou susciter les questions les plus stimulantes ».

À cela nous souhaitons ajouter un septième point qui ne fera sûrement pas l'unanimité parmi les vulgarisateurs mais qui fait partie de nos convictions personnelles : le *parcours initiatique*. Le lecteur d'une revue comme *Quadrature* se considère comme initié en mathématiques, en général, et chaque article qui lui présente un défi de compréhension est pour lui un parcours initiatique particulier,

parcours qui consiste normalement d'une promesse, d'un certain nombre d'obstacles et d'écueils, et d'une révélation finale, en guise de récompense. Ainsi qu'un changement de statut : celui qui a surmonté cette épreuve, celui qui connaît ce secret fait dorénavant partie des initiés et peut-être reconnu par eux comme tel.

Pelay et Mercat [2012] ne parlent pas de parcours initiatique, mais ils définissent une terminologie qui s'en inspire sûrement : dans le diagramme de la figure 4.1 (qui est repris tel quel aussi dans Pelay et Boissière [2015] ; Pelay et Artigue [2016]) ils utilisent les termes *zone magique*, *zone didactique* et *zone maîtrisée*. Les axes représentent les connaissances de l'apprenant et celles du dispositif (le texte, dans notre cas) par ordre de difficulté croissante.

- La *zone magique* est celle où le contenu proposé dépasse significativement les connaissances des lecteurs. Pour citer Pelay et Mercat [2012] : « Nous parlons de « magie » dans le sens où le public n'a aucune prise sur la réalité mathématique qui lui est proposée : elle lui est même invisible, incompréhensible, inaccessible » ;
- la *zone didactique* est celle où une compréhension est possible ;
- la *zone maîtrisée* est celle où le public a une certaine maîtrise du contenu, il peut se « racrocher » à des choses connues.

Le terme de « magie » fait bien sûr référence à la troisième loi de Clarke : « Toute technologie suffisamment avancée est indiscernable de la magie » (*Profiles of the Future*, H.M.H. Publishing, 1958), mais ce qui est sous-étendu dans ce choix terminologique est le fait que la magie a une force d'attraction indéniable et cette force peut être bien utile pour motiver et captiver le lecteur.

Dans la suite de ce chapitre nous allons décrire, à travers deux exemples d'articles, notre approche à la vulgarisation, approche basée sur l'audace et le défi dans le but de générer de l'ambition chez le lecteur, de le captiver et de l'accompagner à travers son parcours initiatique.

Ces articles[1], parus en 2015 et 2019, portaient sur deux sujets classiques mais bien éloignés des mathématiques consensuelles et du programme pédagogique : la sémantique formelle de Montague [YH, 2015] et la logique combinatoire à travers la métaphore des oiseaux de la forêt enchantée de Smullyan [YH, 2019d].

4.2 L'approche formelle de Montague

Richard M. Montague a été un personnage controversé : professeur d'université en linguistique, il n'a pas hésité de défier ses étudiants en proposant une formalisation logique du langage naturel dont la densité de notation avait de quoi choquer un logicien confirmé. Cela est merveilleusement bien décrit par Aifric Campbell dans son roman *The Semantics of Murder* (Serpent's Tail, 2009). La scène se passe un samedi dans un amphithéâtre de l'université UCLA lors d'un séminaire destiné aux linguistes et c'est Robert (le personnage inspiré de Montague) qui donne le cours, alors que son frère Jay y assiste :

[1] Je tiens à remercier le rédacteur-en-chef de la revue Jean-Paul Truc pour sa confiance ainsi que les deux relecteurs, l'un éponyme (Sylvain Kahane) et l'autre anonyme, pour leurs inestimables conseils et suggestions. La relecture de l'article de 2019 par le deuxième relecteur a été une véritable aventure humaine puisque les corrections m'ont été transmises sous forme de 2h30 d'enregistrements sonores et j'ai pu plusieurs fois sentir la voix du relecteur passer de l'éloge (modérée) à l'indignation (totale)... Et puisqu'on en est aux remerciements, je suis reconnaissant à Daniel Lehmann qui au début des années 1980, à l'université de Lille, donnait un cours optionnel de DEUG A 1re année, intitulée « Introduction au raisonnement rigoureux en mathématiques », cours qui m'a marqué à vie.

« Prenons une phrase simple comme “Tout homme parle”. La traduction usuelle en logique serait

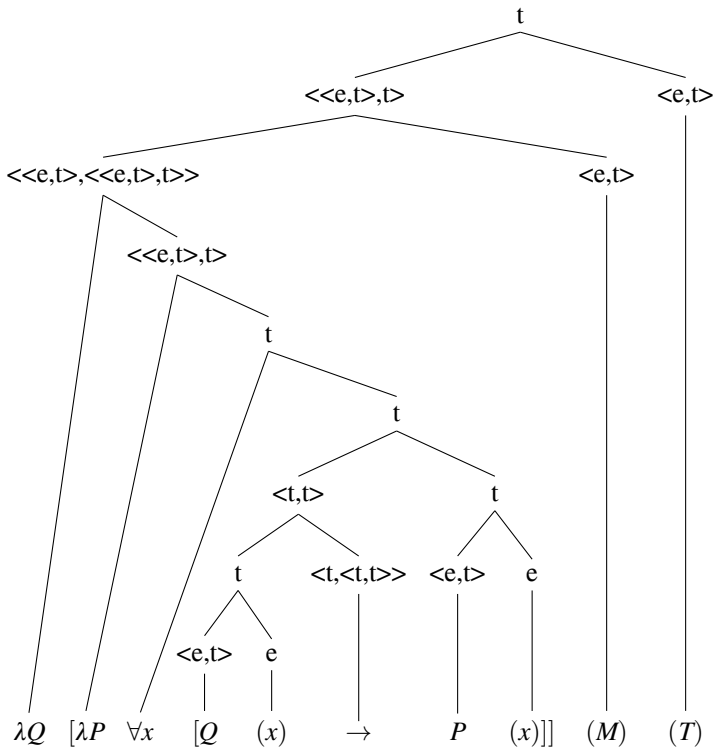
$$\forall X[M(X) \rightarrow T(X)].$$

« Mais ceci est clairement insatisfaisant.» La classe regardait la formule dans un silence intimidé. Une fille assise devant Jay se tourna vers son camarade et haussa les épaules avec un sourire nerveux, montrant un grand point d'interrogation qu'elle avait tracé dans son cahier de cours.

« Et voilà la solution que je vais proposer », continua Robert.

$$\lambda Q[\lambda P \forall X[Q(x) \rightarrow P(x)]](M)(T).$$

Jay regarda les têtes des étudiants qui bougeaient, en grimaçant pendant leurs prises de notes et en comptant diligemment les crochets.



La craie se brisa sur le tableau noir et claqua sur le sol devant les pieds de Robert. La fille devant Jay secoua la tête. Robert tapota sur le lutrin et Jay pouvait sentir l'enthousiasme dégringolant de son frère. Son regard a parcouru l'audience empreint d'une frustration à peine déguisée.

La scène est fictive mais le texte montre bien le décalage entre le professeur formalisant à outrance et le public frustré de linguistes en herbe ne disposant, sans doute, que d'un bagage mathématique minimum. Et l'extravagance de Montague ne s'arrête pas là : il est peut-être le personnage le plus janusien de la linguistique, menant en parallèle une vie d'universitaire, homme d'affaires respecté et organiste de sa paroisse, en même temps qu'une vie d'homosexuel adonné à toutes sortes de plaisirs, plaisirs qui apparemment lui ont été fatals puisqu'il est mort assassiné sous des circonstances non élucidées mais sans doute liées à son côté obscur.

Nous n'aurions pas pu faire de meilleur choix pour un premier article dans *Quadrature*.

4.2.1 Être ambitieux dans ses objectifs

Aux aspects croustillants de la personnalité de Montague s'ajoute l'envergure de sa théorie : si, après lecture du titre « Les mathématiques de la langue », le lecteur de l'article s'attend à trouver quelque statistique sur la fréquence des verbes ou des adjectifs épithètes, il sera bien surpris par l'ambition de la théorie de Montague dont nous définissons le contexte dans la 2e section de l'article :

> La langue sert avant tout à parler du monde. Nos points de départ seront donc le « monde » et la « langue ». Le « monde », que nous noterons $\mathfrak{Monde}$, peut être le monde réel, un monde imaginaire, ou simplement un ensemble d'objets ou d'idées abstraites.

Le lecteur-type de *Quadrature* est habitué à voir divers phénomènes du monde réel modélisés mathématiquement, selon le bon vieux principe de Galilée « la nature est écrite en langage mathématique ». Mais ici on plane bien plus haut : nous utilisons un simple symbole ($\mathfrak{Monde}$) pour représenter *le monde tout entier*. L'utilisation de caractères gothiques et du mot dans sa totalité (et non pas d'une lettre seule, comme il est coutume en mathématiques) accentue l'étrangeté de cette notation[2]. À peine dépassée la première page de l'article, le lecteur se trouve devant une découverte : dans une petite formule toute simple il trouve réunis trois ensembles tout aussi énormes que différents : (a) le monde dans lequel il vit (ou celui qu'il imagine), (b) l'ensemble de toutes les « phrases françaises, orthographiquement et syntaxiquement correctes, qui se réfèrent aux objets de l'ensemble $\mathfrak{Monde}$ », et en position intermédiaire, (c) un ensemble de formules écrites dans des formalismes que l'article se propose de lui faire découvrir :

> On a donc la situation suivante :
>
> $$\mathfrak{LangTexte} \xrightarrow{\textsf{Analyse}} \mathfrak{LangLog} \xrightarrow{\textsf{Inter}} \mathfrak{Monde},$$
>
> où **Analyse** est l'analyse qui nous permet de représenter les phrases de texte en langage intermédiaire, et **Inter** le lien référentiel entre $\mathfrak{LangLog}$ et $\mathfrak{Monde}$. On appelle cette dernière opération, *interprétation*[3].

On est là en pleine zone magique : le lecteur se retrouve non pas devant un petit fragment de la réalité mais devant sa totalité. L'ambition dans la détermination des objectifs à atteindre est primordiale à ce stade, juste avant le traditionnel « Dans cet article nous allons...», puisqu'il y va de la décision du lecteur de poursuivre ou non la lecture de l'article. Nous espérons que la petite phrase

> Le « monde », que nous noterons $\mathfrak{Monde}$ peut être le monde réel...

aiguillera sa curiosité et lui donnera l'afflux d'énergie nécessaire à la confrontation avec le contenu mathématique qui suit.

4.2.2 Mathématiser le langage naturel

La section suivante traite de la mathématisation du langage naturel. On aurait pu commencer par un exemple simple : l'arbre syntaxique d'une phrase de type sujet-verbe-complément, sa traduction en un prédicat binaire. Mais ce serait passer à côté du plaisir esthétique (pour ne pas dire hédonique) procuré par l'approche bourbakiste qui consiste à définir d'abord un cas général en toute abstraction

[2] Sans parler de la référence implicite à la typographie du logo du journal *Le Monde*...

[3] Attention, ici le mot « interprétation » a un sens technique strict, qu'il ne faut pas confondre avec celui du langage courant.

pour passer ensuite au cas particulier, et enfin, en dernière instance, à l'exemple simple et utile. Et ce plaisir esthétique du texte serait inopérant si sa compréhension n'était pas un obstacle, un cap à passer, une étape du parcours initiatique.

Pour faire le lien entre la langue (objet par définition ambigu et impalpable) et les mathématiques (dans toute leur clarté cristalline), le premier résultat de l'article n'est pas un théorème mais un principe (donc, intuitivement, une vérité générale que l'on ne se donne pas la peine de démontrer) :

> **Principe de compositionnalité (Frege).** La sémantique d'une phrase s'obtient à partir des sémantiques de ses parties et de la manière dont elles ont été composées (= la syntaxe de la phrase).

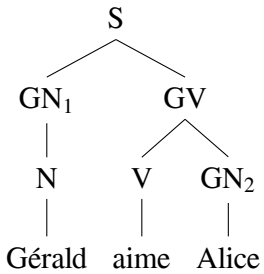
Ce principe justifie la distinction épistémologique entre syntaxe et sémantique et trace l'itinéraire de l'article : on analysera d'abord syntaxiquement les données linguistiques et ensuite on se servira de cette analyse pour accéder au sens.

Suivent des définitions purement mathématiques des notions de *monoïde libre* Σ^* sur un ensemble Σ (qui, accessoirement, s'appelle « alphabet »), et de *langage formel* comme étant un sous-ensemble *quelconque* de Σ^*. L'accent est ici donné à l'adjectif « quelconque ». En effet, la définition

> Un **langage formel** L sur un alphabet Σ est un sous-ensemble (quelconque) de Σ^*

a cela de rare que l'on y donne un terme savant (« langage formel »), prometteur de structure complexe et sophistiquée, en totale contradiction avec la banalité de la caractérisation « sous-ensemble quelconque ». Pour passer du chaos à l'ordre, de la généralité totale à la spécificité, nous arrivons aux notions de *grammaire formelle (hors contexte)* et *langage formel engendré par une grammaire*. Semée d'embûches, cette série de définitions se termine par un exemple simple : une grammaire formelle qui reconnaît la phrase « Gérald dort », comportant cinq règles de production. Si les définitions se trouvent dans la zone magique et la phrase simplissime « Gérald dort » dans la zone maîtrisée, c'est pour inciter le lecteur à « démagiser » cette partie de connaissances et la faire entrer dans sa zone didactique.

Et après cet exemple simple de règles de production, un exemple d'arbre syntaxique utilisant les notions de groupe nominal et de groupe verbal, auxquelles le lecteur de l'article ne peut être que familier :



Nous avons pu, à multiples occasions, faire l'expérience de présentation de tels arbres syntaxiques à des collégiens (cf. § 4.2.7), sans qu'il y ait jamais le moindre signe d'incompréhension ou de malaise devant le formalisme, nous sommes donc convaincus que même si les arbres syntaxiques ne font pas partie de la transposition didactique consensuelle, leur assimilation ne pose aucun problème

significatif[4]. Entre définitions très abstraites et exemples linguistiques très concrets, le lecteur doit normalement, à ce stade, avoir le sentiment d'avoir maîtrisé le premier stade de l'initiation.

4.2.3 Logique et interprétation

Suit une nouvelle série de définitions, celle de *formule logique* (de la logique du 1er ordre) et d'*interprétation* de formule logique. Dans un cours classique on parlerait d'abord de théorie abstraite (en s'efforçant de ne pas parler de *vérité*, alors que c'est cela-même qui permet de distinguer au mieux les fonctions des prédicats) pour passer ensuite à la théorie des modèles. Dans notre cas, tout arrive naturellement puisqu'on a déjà positionné la logique en tant que « langage intermédiaire », entre le langage naturel et le « monde ».

Dans un paragraphe comme

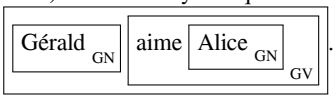
> Une phrase du type ***Gérald dort*** décrit une situation où il y a un agent identifié par le nom ***Gérald*** qui effectue l'action de dormir. Dans la formule logique il est naturel de prendre une constante g pour représenter ***Gérald***, qui ensuite sera envoyée par **Inter** à l'entité du monde réel qui correspond à ***Gérald*** — notons cette entité **gérald**. On a donc $\mathbf{Analyse}(\textbf{\textit{Gérald}}) = g$ et $\mathbf{Inter}(g) = \mathbf{gérald}$,

on arrive à distinguer les trois facettes de Gérald (le mot, la constante logique, le référent dans le monde réel) par des truchements typographiques (bâton italique gras, initiale seule en italiques maigres, gras droit). Par cet effet (qui relève de la *mise en scène* selon Rittaud [2015]) le lecteur comprend qu'on a trois « mondes » (la langue, la logique, le monde réel) mais qu'on parle tout de même « de la même chose » : notre compréhension humaine fait le pont entre ces trois « mondes » et les deux applications « Analyse » et « Interprétation » nous permettent de faire ces liens indispensables entre mots, objets mathématiques et objets réels.

L'interprétation ensembliste de la notion de prédicat permet de consolider les notions présentées par un simple exemple : si « dort » est un prédicat unaire, son interprétation $\mathbf{Inter}(\text{dort})$ peut être considérée comme un sous-ensemble du domaine et l'interprétation de l'application du prédicat à la constante « g », $\mathbf{Inter}(\text{dort}(g)) = \mathbf{Inter}(\text{dort})(g)$ devient alors la réponse à la question « g appartient-il à $\mathbf{Inter}(\text{dort})$? ».

Une fois la traduction du langage naturel effleurée et l'interprétation ensembliste du formalisme logique amorcée, on arrive à un autre cap. C'est le moment où il faut expliquer que si on s'intéresse aux arbres syntaxiques, c'est parce que tous les nœuds sont importants, et non pas uniquement les feuilles :

> Montague aurait pu s'arrêter là, en disant : « je sais représenter mes terminaux (***Gérald***, ***aime***, ***Alice***) et mon axiome de départ (S) en langage logique, peu me chaut le reste ».
>
> Que nenni ! Son génie a consisté — entre autres — à dire que si l'on veut être honnête avec soi-même, si l'on veut aller au fond des choses, alors *le principe de compositionnalité doit s'appliquer partout*, aussi bien dans $\mathfrak{LangTexte}$, que dans $\mathfrak{LangLog}$, et la représentation en logique d'une composition doit être la composition des représentations.

[4] Il paraît que les arbres syntaxiques ne sont pas (plus ?) enseignées au secondaire, et ceci malgré les recommandations de nos amis Québecois [Bélanger et Gauvin, 2017]. Néanmoins, d'après ma fille Danaé (lycéenne) la structure syntaxique est bien représentée au lycée, non pas sous forme d'arbre mais sous forme de boîtes imbriquées : 

.

4.2.4 Théorie des types et λ-calcul

Si on la considère du point de vue du contenu et de l'enchaînement logique des notions, la théorie des types n'est pas absolument indispensable dans notre texte : on aurait pu passer immédiatement au λ-calcul, ce qu'on fera une page plus loin. Mais la théorie des types a l'avantage de montrer trois choses : primo que les deux types primitifs *e* et *t* se composent pour produire une infinité de types complexes ; secundo que chaque feuille d'un arbre syntaxique peut être de type différent et que la fréquence d'un mot peut être inversement proportionnelle à la complexité de son type ; et tertio qu'à la manière d'un puzzle, les types d'un arbre interagissent et la moindre erreur de type rend l'assemblage de l'arbre syntaxique impossible.

De ce fait, les types constituent une étape intermédiaire qui préfigure la complexité de la représentation des mots en λ-calcul. La définition de la notion de λ-opérateur est un véritable défi didactique. La cause en est sans doute le flou terminologique qui règne autour de l'appellation des fonctions dans l'enseignement des mathématiques. Quand on écrit « $\sin(\frac{\pi}{2})$ », il est clair qu'il s'agit d'une valeur (à l'occurrence, 1), et quand on écrit « sin », il est clair qu'il s'agit d'une fonction. Mais qu'en est-il de l'écriture « $\sin(x)$ » ? S'agit-il de la fonction qui à une valeur quelconque de x associe une valeur ? Ou de la valeur de la fonction sinus pour un x qui a une valeur donnée, mais qui nous est inconnue ?

Nous avons relevé le défi par l'étalage d'une liste d'exemples de difficulté de compréhension croissante (pour finir avec un exemple trivial) :

> Ainsi, $\lambda x.f(x)$ est la même chose que $x \mapsto f(x)$, c'est-à-dire la fonction f. L'intérêt de la notation est qu'elle nous permet de définir toutes sortes de fonctions. Par exemple, $\lambda x.\lambda y.(x+y)$ est la fonction (de deux variables) qui à (x, y) associe $x + y$, alors que $\lambda x.(x + y)$ est la fonction (d'une variable) qui à x associe la somme $x + y$ (sans donner plus d'information sur y). De même, $\lambda P.P(x)$ est la fonction qui associe à un prédicat unaire sa valeur en x et $\lambda P.\lambda x.P(x)$ est la fonction qui à P et à x associe $P(x)$. Les amateurs de λ-calcul s'amusent même à noter $\lambda x.x$ la fonction identité.
>
> Une fonction f peut être appliquée à une valeur x, on note le résultat $f(x)$. De la même manière, on peut *appliquer* une λ-expression à une valeur. Ainsi $(\lambda x.\sin(x))(\frac{\pi}{2})$ est tout simplement $\sin(\frac{\pi}{2})$. On appelle cela, tout naturellement, une *application*,

qui nous ramène à une notion bien connue du lecteur (mais qui sera ébranlée dans l'article suivant, lorsqu'on parlera de logique combinatoire cf. § 4.3.2) : celle d'*application* d'une fonction à une valeur. On est donc de nouveau en zone maîtrisée avec comme mission de ramener ce qui a été dit en zone magique dans l'espace de la zone didactique.

4.2.5 Calculs et phénomènes grammaticaux

Les quatre pages suivantes de l'article sont des pages de calcul. Cela est rassurant pour le lecteur qui va suivre pas-à-pas les calculs et améliorera ainsi sa compréhension du sujet, mais aussi pour le lecteur qui les survolera rapidement, considérant que ce qui importe le plus ce ne sont les calculs eux-mêmes, mais les résultats.

Les résultats en question correspondent à des phénomènes grammaticaux (toujours en zone maîtrisée) : la *coordination* (« Gérald aime Alice mais préfère Alexia »), la *quantification* (« Tout le monde aime Alice ») et, enfin, l'*article défini* (« Le philosophe aime Alice »).

Mine de rien, ce dernier phénomène permet la (ré-)introduction et (re-)définition d'une notion fondamentale des mathématiques : celle de l'*égalité*. Moment magique dans l'enseignement de la logique, et constamment renouvelé lors de nos cours, celui de la question posée aux élèves de 3^{e} année

d'école d'ingénieurs : « à votre avis, que signifie $a = b$? ». Il s'agit certainement d'un mythe, mais ne dit-on pas que les premières choses qu'on apprend à l'école sont le « b-a-ba » (articulation des phonèmes ou concaténation en langage formel) et le « $1 + 1 = 2$ » (addition des unités multiplicatives de $\mathbb{N}$ et égalité) ? Interpréter l'égalité comme une contrainte posée sur toutes les interprétations possibles d'une formule donnée ne peut qu'ouvrir l'esprit et faciliter la compréhension de la différence entre théorie abstraite et interprétation.

En plus, cela permet de comprendre la notion d'*unicité* : l'exemple donné dans l'article « il n'existe qu'un seul philosophe quelque soit l'interprétation » est formalisé par $\exists x(\forall y(\mathrm{philosophe}(y) \leftrightarrow x = y)$, ce qui montre aussi le luxe de concision que constitue la notation « $\exists!$ » qu'on utilise couramment en mathématiques, mais jamais en logique. Le corps de l'article se termine donc sur le paradoxe qui caractérise l'approche de Montague : l'article défini, un des mots les plus fréquents de la langue française, a comme formalisation la formule impitoyable

$$\lambda Q(\lambda P(\exists x\,(\forall y\,(Q(y) \leftrightarrow (x = y)) \wedge P(x)))).$$

Le parcours initiatique se termine, le lecteur capable de comprendre le sens et la nécessité de cette formule fait désormais partie des « initiés », sa persévérance a été récompensée. Et il a appris (ou du moins s'est aperçu de) cette « loi de Zipf inversée » de la sémantique formelle de Montague : la loi de Zipf dit que les mots les plus fréquents sont les plus courts, ici la loi de Zipf inversée fait que les mots les plus fréquents ont la formalisation logique la plus complexe.

4.2.6 Épilogue

L'épilogue, qui dans un ouvrage scientifique fait plutôt office de synthèse et de consolidation, est dans un article de vulgarisation l'opportunité de changer de rythme et d'accentuer le côté *mise en perspective* de Rittaud [2015] : la prudence des explications laisse sa place à la richesse des pointeurs vers des notions et des domaines insoupçonnés. Ainsi nous mentionnons brièvement la notion d'*inférence* qui permet de définir les notions de *théorie* et de *théorème* (et donne ainsi un sens à ces deux termes qui font partie de la zone maîtrisée du lecteur), nous nous posons la question de la *temporalité*, nous parlons de la logique *modale* qui introduit les opérateurs « nécessairement » et « peut-être », nous introduisons les notions de descriptions *intensionnelle*[5] et *extensionnelle* et nous finissons triomphalement avec la citation maintes fois évoquée de Wittgestein, que « les limites de ma langue sont les limites de mon monde », citation qui ne peut que faire vibrer la corde sensible de tout jeune lecteur qui pense que, tel le Pays Fantastique de Michael Ende, son monde n'a pas de limite...

4.2.7 Parenthèse : présenter la sémantique formelle de Montague à des collégiens

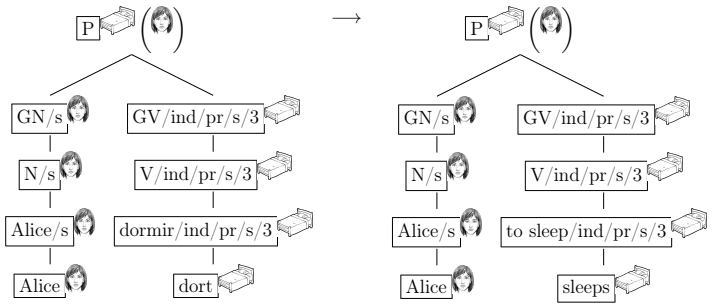
Dans le cadre des journées « Le numérique, des métiers en tous genres » organisée par plusieurs établissements d'enseignement supérieurs bretons ces dernières années, nous avons organisé à plusieurs reprises des ateliers d'informatique déconnectée, sur le thème de la sémantique formelle de Montague, destinés à des groupes de 10–15 collégiens. Le thème annoncé de cet atelier de 45′ était « Comment l'ordinateur gère les langues et les langages », thème qui contient déjà la promesse implicite de clarifier la différence entre « langue » et « langage ».

[5] Avec le petit clin d'œil orthographique subversif de la graphie « intension ».

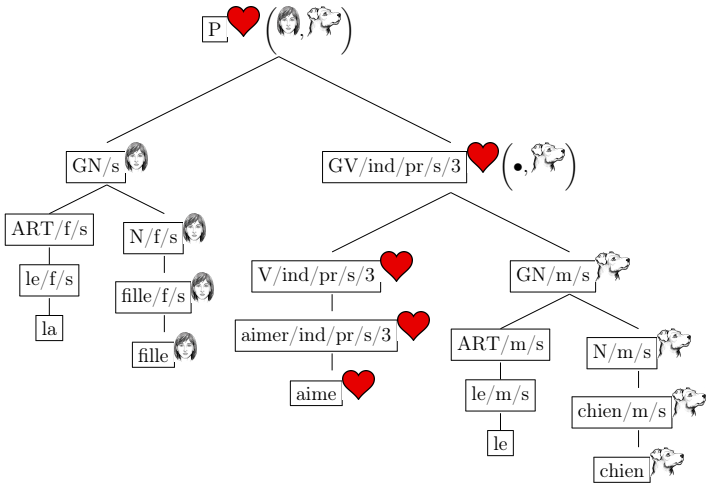
L'atelier commence par la construction progressive au tableau (par un échange constant avec les élèves) de l'arbre syntaxique de la phrase « Alice dort », en commençant par les feuilles. Ensuite, pour parler de « sens », nous avons préparé des images magnétiques représentant une jeune fille (« Alice ») et un lit (l'action de dormir), en plaçant ces images à côté des feuilles de l'arbre syntaxique nous prétendons que c'est le « sens » des mots, ou du moins une représentation de celui-ci. En passant du statut symbolique (dans la trichotomie de l'objet selon Peirce) au statut iconique, nous donnons l'illusion d'une meilleure représentation du sens, tout à fait idoine à un public familier des émoticônes.

Moment critique de l'atelier : le moment où il y a une composition à faire, lorsqu'on passe de « GN GV » à « P ». « Suffit-il de mettre l'image d'Alice à côté de celle du lit pour dire qu'Alice dort ? Et si oui, dans quel ordre ? N'oublions pas que l'image du lit symbolise l'action de dormir, comme dire qu'Alice est l'agent du verbe dormir ? » Dans certains cas les élèves ont spontanément proposé de mettre des parenthèses, dans d'autres cas nous le leur avons suggéré, mais cela ne les a jamais choqués.

Ainsi nous avons découvert tout naturellement la notion de prédicat unaire et d'argument de prédicat. Mais nous ne nous arrêtons pas là. Nous demandons aux élèves de choisir une langue (c'est tantôt l'anglais, tantôt l'espagnol) et nous traçons un deuxième arbre syntaxique, cette fois de la racine vers les feuilles, pour aboutir à cette autre langue. Voici le résultat du traitement de cette première phrase :



La zone magique est ici le traitement de la langue par l'ordinateur, la zone maîtrisée comporte les notions de grammaire (et de langue étrangère) que les élèves possèdent en arrivant. L'atelier se poursuit avec la phrase « Alice aime le chien », dont l'arbre syntaxique est représenté comme suit :



Cela permet d'introduire tout naturellement la notion de prédicat binaire (avec l'ordre des arguments) qui fait donc partie de la zone didactique, tout en laissant dans la zone magique le • qui

apparaît dans la formalisation du groupe verbal (et qui ne peut être formalisé décemment que par une λ-expression, difficile à expliquer à des élèves qui n'ont pas encore appris la notion de fonction).

L'atelier se termine par l'arbre syntaxique de l'expression en langage de calculatrice « $(3 \times 7) + (5 \times 6)$ » où les élèves font le parallèle entre le sens d'un langage naturel et celui d'un langage de programmation, à l'occurrence le langage des quatre opérations arithmétiques, où le « sens » d'une expression est la valeur numérique obtenue en faisant les calculs. Dernier défi laissé aux élèves : quel est le sens du mot « sens » lorsque dans un cas il s'agit de situations de la vie de tous les jours et de l'autre d'une simple valeur numérique ?

Après cette variation didactique sur le thème de sémantique formelle de Montague, revenons à nos articles dans *Quadrature*.

4.3 La logique combinatoire

Le deuxième article [YH, 2019d] prend comme point de départ la nouvelle *Ne vous moquez pas de l'oiseau moqueur* (titre qui se moque, bien sûr, du roman *Ne tirez pas sur l'oiseau moqueur*) de Raymond Smullyan [1985], mathématicien célèbre pour ses casse-têtes mathématiques (réunis en France dans le volume de 720 pages *Soyons fous !*, Dunod 2007) et l'excellente pédagogie de ses ouvrages d'introduction à la logique.

Smullyan a relevé le défi de présenter la logique combinatoire, discipline austère et difficile d'accès, par une métaphore surprenante : celle d'oiseaux doués de parole dans une « certaine forêt enchantée » : *A certain enchanted forest is inhabited by talking birds...* est l'incipit de cette nouvelle de 70 pages, qui n'a (malheureusement) pas encore été traduite en français.

L'article [YH, 2019d] présente des défis bien plus importants que le premier, aussi bien pour le lecteur que pour son auteur. Notons que la revue *Quadrature* nous a fait l'honneur de décorer la couverture du fascicule (cf. figure 4.2) par un bel oiseau moqueur, en hommage aux oiseaux de Smullyan.

FIG. 4.2: Couverture du fascicule 113 (2019) de la revue *Quadrature*.

4.3.1 Fonctions primitives récursives

Le monde imaginaire de Smullyan est très attirant et on serait tenté de s'y plonger immédiatement pour commencer l'article, mais cela n'aurait comme résultat que d'augmenter le gap entre cette belle métaphore aviaire et les potentielles applications concrètes. Nous avons donc décidé de commencer par un sujet en apparence totalement décorrélé et potentiellement surprenant pour les lecteurs, les *fonctions primitives récursives* :

> Nous allons nous intéresser à un ensemble particulier de fonctions, celles qui sont *calculables par une procédure effective*, autrement dit : *dont un programme d'ordinateur peut calculer les valeurs exactes*. On se restreint à celles dont le domaine de définition est un sous-ensemble de $\mathbb{N}^k$ et qui sont à valeurs dans $\mathbb{N}$.
>
> Une **fonction primitive récursive de base** peut être de trois types :
>
> 1. la fonction constante 0 ;
> 2. la fonction *successeur* σ ;
> 3. une fonction de *projection* π^i_k.
>
> La fonction *successeur* envoie tout entier naturel n à $n+1$. La fonction de projection π^i_k envoie un k-uplet d'entiers vers le i-ème entier (en particulier, π^1_1 est l'identité, $\pi^1_2(x,y) = x$ et $\pi^2_2(x,y) = y$ pour tous x, y).

La définition suivante est celle de la *récursion primitive* : si g est une fonction $\mathbb{N}^k \to \mathbb{N}$ et h est une fonction $\mathbb{N}^{k+2} \to \mathbb{N}$, alors la fonction $f : \mathbb{N}^{k+1} \to \mathbb{N}$ est *définie à partir de g et h par récursion primitive*, si et seulement si $f(n_1,\ldots,n_k,0) = g(n_1,\ldots,n_k)$ et $f(n_1,\ldots,n_k,\sigma(m)) = h(n_1,\ldots,n_k,m,f(n_1,\ldots,n_k,m))$ où σ est la fonction successeur.

Il s'en suit tout naturellement la définition de *fonction primitive récursive* comme étant une fonction primitive récursive de base, ou une fonction obtenue à partir d'un nombre quelconque de compositions et de récursions primitives appliquées aux fonctions primitives récursives de base.

Nous atteignons l'endroit critique où le lecteur va décider s'il poursuivra ou non la lecture de l'article avec un exemple qui ne peut qu'intriguer le lecteur et une promesse qui réunit de manière surprenante trois disciplines inattendues :

> La fonction $f(n) = 3n+2$ est donc une fonction primitive récursive, puisqu'elle peut s'écrire
>
> $$f(n) = \text{somme}(\text{produit}(\sigma(\sigma(\sigma(0))), n), \sigma(\sigma(0))).$$
>
> Dans la suite, quand nous parlerons de « fonction » il sera entendu « fonction primitive récursive ». Nous verrons à la section VI que les objets de la logique combinatoire, appelés *combinateurs*, représentent ce type de fonctions : à chaque fonction on peut associer un combinateur, qui devient un « programme » si on considère la logique combinatoire en tant que langage de programmation,

les trois disciplines étant donc l'analyse (puisqu'on parle de fonctions), la logique et l'informatique (puisqu'on parle de langage de programmation).

4.3.2 « Elle nous jette hors de toutes nos habitudes de pensée... »

Présenter les principes de base du formalisme de la logique combinatoire comporte un défi de taille. C'est sans doute pour cette raison qu'un logicien averti, Robert Feys, écrivait en 1946 [Feys, 1946] :

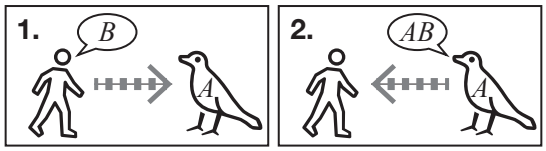
> reconnaissons qu'aucune partie de la logique formalisée ne présente un premier abord aussi insolite et aussi rebutant [que la logique combinatoire]. [Elle] ne dérange pas seulement nos habitudes d'écriture ; elle nous jette hors de toutes nos habitudes de pensée.

Quel est le problème ? En logique combinatoire on peut appliquer n'importe quel élément d'un ensemble à n'importe quel autre élément du même ensemble. Fini la séparation claire et nette entre « fonction » et « variable » : à côté du bien connu $f(x)$ on peut tout aussi bien écrire $x(f)$, $f(f)$ et même $x(x)$... On dénote par (AB) l'*application de A à B* et cette opération n'est ni commutative, ni associative. De plus, pour réduire l'utilisation de parenthèses au strict minimum, on adopte une convention de priorité à gauche : $fxyz$ signifie « f appliqué à x, le résultat appliqué à y, le résultat appliqué à z », c'est-à-dire $(((fx)y)z)$, et $fx(yz)$ signifie $((fx)(yz))$.

Comment faire passer ce cap au lecteur ? Nous avons (naïvement, peut-être) pris le parti audacieux d'espérer que la citation de Feys, introduite explicitement dans le texte, conduirait le lecteur à relever le défi. Il ne s'agit plus de simple zone magique, mais de zone magique *dangereuse* puisqu'elle a le pouvoir de nous déséquilibrer en « nous jetant hors de toutes nos habitudes de pensée »...

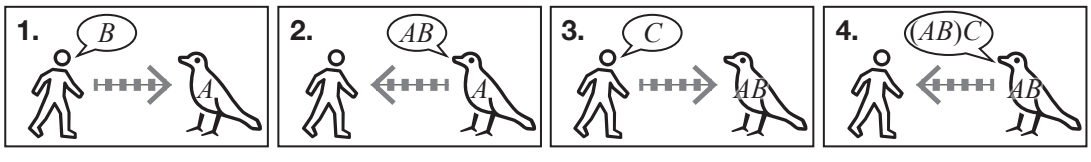
4.3.3 Les oiseaux facilitent-ils la compréhension ?

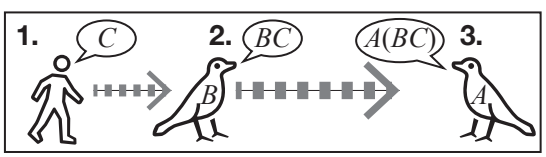
Dans la métaphore de Smullyan les oiseaux parlent, certes, mais leur vocabulaire est très réduit : il se limite aux noms d'oiseaux. Nous avons la situation suivante : lorsqu'on s'adresse à un oiseau A en disant B, il répond par AB



Cette opération fictive présente un souci didactique : il faut éviter de confondre *logogramme* (symbole de variable mathématique) et *phonographème* (représentation graphique d'une chaîne de phonèmes, humains ou aviaires). En effet, le lecteur doit garder en tête le fait que le personnage qui se promène dans la forêt enchantée ne prononce pas « B » mais un nom d'oiseau symbolisé par la variable B et que de même l'oiseau A ne répond pas par « AB » ni même par une concaténation des noms symbolisés par les variables A et B, mais par un tout autre nom qui est symbolisé par AB (l'*application* de A à B). Ce qui dérange dans cette métaphore c'est l'absence totale de lien entre signifiant et signifié : le fait de nommer cet objet AB ne nous apprend rien sur le nom qu'il représente, et il n'y a aucun lien entre ce nom et les noms représentés par A et par B.

Pour éviter au lecteur de tomber dans ce piège de compréhension, nous avons créé deux autres « bandes dessinées » qui représentent la manière d'obtenir $(AB)C$ et $A(BC)$ et illustrent donc le défaut d'associativité de l'opération d'application :





en espérant qu'elles faciliteront la compréhension de la métaphore de Smullyan.

4.3.4 Texte mathématique à deux niveaux

Un autre effet de *mise en scène* selon Rittaud [2015], pour avoir le beurre *et* l'argent du beurre, c'est-à-dire le texte métaphorique de Smullyan *et* son explication mathématique, nous avons utilisé, dans cette section uniquement, deux types de texte. Le premier, en caractère romain, décrit la métaphore de Smullyan, et le deuxième, en caractère bâton, fournit des explications mathématiques. En voici un exemple :

> Et puisqu'on parle de composition, Smullyan introduit la notion suivante :
>
> **Définition 5.** On dit qu'un oiseau M *compose* un oiseau A avec un oiseau B, si pour tout oiseau x on a la condition $Mx = A(Bx)$.
>
> Autrement dit : quand on s'adresse à M on obtient les mêmes réponses que comme si on s'adressait à B qui parle à A, qui nous répond.
>
> LA NOTION DE COMPOSITION MÉRITE QUELQUES EXPLICATIONS. DANS L'UNIVERS DES FONCTIONS DE L'ENSEIGNEMENT SECONDAIRE OU DU PREMIER CYCLE D'ÉTUDES, ON A LE GRAND LUXE DE ***TOUJOURS*** POUVOIR COMPOSER DES FONCTIONS f ET g — À CONDITION BIEN SÛR QUE L'IMAGE DE LA PREMIÈRE SOIT COMPRISE DANS LE DOMAINE DE DÉFINITION DE LA DEUXIÈME. QUAND CETTE CONDITION EST ASSURÉE, $f \circ g$ EXISTE TOUJOURS EN TANT QUE FONCTION. CE N'EST PAS LE CAS DANS LA FORÊT ENCHANTÉE : ON PEUT, POUR TOUS A, B ET x, ÉCRIRE $A(Bx)$ (LES PARENTHÈSES SONT ICI PRIMORDIALES), ET CE TERME FAIT ÉVIDEMMENT PARTIE DE L'ENSEMBLE. MAIS CELA NE VEUT NULLEMENT DIRE QU'IL EXISTE UN AUTRE ÉLÉMENT M TEL QUE $Mx = A(Bx)$ POUR TOUT x, DONC LA « COMPOSÉE » DE A ET DE B N'EXISTE PAS FORCÉMENT DANS UN ENSEMBLE QUELCONQUE E.

En exposant le lecteur en même temps à la métaphore originale des oiseaux et à son exégèse, en le confrontant à l'imaginaire ludique et à la rigueur mathématique, nous espérons maximiser l'apport de ce texte de Smullyan.

Au fil du texte, la métaphore de Smullyan évolue. S'il y a quelques embûches au début, les deux fils de texte (métaphore et exégèse) se rapprochent de plus en plus par la suite. Ainsi nous définissons l'identité (lettre **I**), les *répétiteurs* (lettre **M**, l'oiseau moqueur et **W**, le weka), les *élimininateurs* (lettre **K**, le kamichi), le combinateur **L** (le loriot[6]), les *compositeurs* (lettre **B**, l'oiseau bleu), les *permutateurs* (lettre **C**, le cardinal, lettre **T**, le tarin et lettre **R**, le rouge-gorge) et, enfin, lettre **S**, le sterne, qui est répétiteur, permutateur et compositeur en même temps.

Suivent deux notions de préparation au résultat fondamental : un oiseau A est appelé *combinatoire d'ordre n* si n est le plus petit nombre tel que pour toute suite d'oiseaux $(X_1, \dots, X_n)$, l'expression $AX_1 \cdots X_n$ puisse se réécrire en n'utilisant que des éléments de l'ensemble $\{X_1, \dots, X_n\}$. Exemple : **B** est d'ordre 3 puisque, par définition $\mathbf{B}xyz := x(yz)$ pour tout x, y, z. Et également le fait qu'un oiseau est *dérivable* d'un ensemble d'oiseaux s'il peut se réécrire en n'utilisant que des éléments de cet ensemble. Exemple **S** est dérivable de {**B**, **C**, **W**} puisque $\mathbf{S}xyz = \mathbf{B}(\mathbf{BW})(\mathbf{BBC})xyz$ pour tout x, y, z.

On finit la section dédiée aux oiseaux avec le résultat fondamental : *tout oiseau combinatoire est dérivable de* {**S**, **K**} *ainsi que de* {**I**, **B**, **C**, **W**, **K**}.

4.3.5 Retour à la rigueur et applications

À ce stade nous abandonnons définitivement le principe du texte à deux niveaux et nous reprenons la théorie des combinateurs de manière rigoureuse, pour arriver au même résultat : {**S**, **K**} et {**I**, **B**, **C**, **W**, **K**} sont des *bases combinatoirement complètes*. Dans cette partie de l'article, les énoncés sont

[6]Nous avons choisi le loriot pour la lettre **L** en hommage à Yvonne Loriod, épouse d'Olivier Messiaen, grand amateur d'oiseaux, qui a créé son *Catalogue d'oiseaux* (œuvre belle et déroutante, dont le deuxième mouvement est justement intitulé *Le loriot*) le 15 avril 1959 à la salle Gaveau.

plus rudes mais ayant gardé les mêmes notations que dans la partie précédente, le lecteur devrait être en mesure de franchir le pas et de comprendre les nouvelles notions introduites.

Puis, fidèles à notre promesse, nous enchaînons avec une section sur les fonctions primitives récursives : la théorie des combinateurs est bien belle, mais comment peut-elle nous servir pour obtenir *toute* fonction primitive récursive ? Nous donnons les combinateurs qui correspondent aux constantes numériques, à la fonction successeur, aux différentes projections, à la composition et à la récursion.

Et comme ce résultat de correspondance entre fonctions primitives récursives et combinateurs est finalement assez abstrait, nous donnons une application des combinateurs : la traduction en logique combinatoire de l'*algèbre de Boole* c'est-à-dire des calculs impliquant des variables binaires et les opérateurs de négation, conjonction et disjonction. Résultat surprenant : les traductions en logique combinatoire des valeurs « vrai » et « faux » sont **K** et **KI** resp., et ce sont les mêmes combinateurs qui représentent les constantes 1 et 0 dans la correspondance entre fonctions primitives récursives et combinateurs.

4.3.6 Application au traitement automatique de la langue

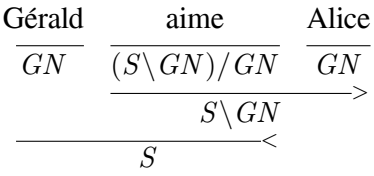
La section VIII relève le défi de donner, sur deux pages, un aperçu de la théorie des *grammaires catégorielles combinatoires* de Steedman [2000] (cf. aussi [Lê Ngoc et YH, 2019] et [Lê Ngoc et al., 2019]).

Première remarque : dans l'article de *Quadrature* précédent nous avons triché en laissant entendre que quelque soit l'ordre des parties du discours (par exemple, sujet en premier, puis verbe, puis complément) l'arbre des λ-expressions serait le même. Il est évident que la traduction de la phrase « Gérald dort » en logique sera toujours « dort(Gérald) », quelque soit la position relative du sujet vis-à-vis du verbe (en arabe le verbe viendrait plutôt en premier). Mais cela ne veut pas dire que les compositions effectuées au niveau de chaque nœud de l'arbre syntaxique sont les mêmes. Normalement la composition de λ-expressions dépend de l'ordre des mots : si « Gérald », représenté par la constante Gérald, vient avant « dort », représenté par $\lambda x.\mathrm{dort}(x)$, on est face à un problème puisqu'on ne peut pas appliquer une constante à une λ-expression. La solution consiste à effectuer une *montée de type*, c'est-à-dire de remplacer la constante Gérald par la λ-expression $\lambda P.P(\text{Gérald})$ (intuitivement : on remplace une constante dont le référent est un objet du monde réel par une construction qui fournit une réponse à toute question sur ce référent). Pour éviter de parler de montée de type nous avons passé sous silence ce problème dans l'article [YH, 2015].

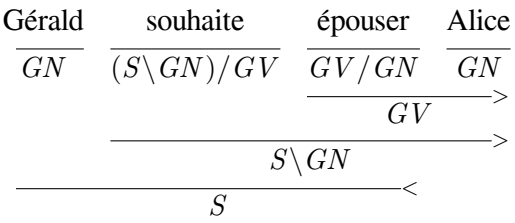
Dans cet article-ci nous considérons le précédent comme acquis et présentons une approche qui donne une solution très élégante au problème de l'ordre des parties du discours, comme aussi à bien d'autres problèmes : l'approche des *grammaires catégorielles combinatoires*. Celle-ci est basée sur l'utilisation de types complexes (appelées *catégories*) qui décrivent *complètement* le comportement syntaxique d'un mot : au lieu d'annoter le verbe « aime » de la phrase « Gérald aime Alice » par un *V* (verbe) on utilisera $(S\backslash GN)/GN$, qui peut s'interpréter par : ce verbe s'applique à un groupe nominal qui se trouve à sa droite et fournit un type $S\backslash GN$ qui, à son tour, s'applique à un groupe nominal qui se trouve à sa gauche, et le résultat de cette opération est une phrase *S*. Il y a donc dans la catégorie que l'on associe à un mot tout le programme de ses interactions avec les autres mots. Ce qui signifie que pour un même mot on peut avoir plusieurs catégories : dans la phrase « Gérald aime » où « aime » n'a pas de COD, on utilisera plutôt $S\backslash GN$ pour le mot « aime ». Et s'il s'agit d'une langue où la position des parties du discours est libre (comme le grec, qui lève les ambiguïtés par l'utilisation de cas) qu'à cela ne tienne : on peut avoir systématiquement plusieurs catégories pour chaque mot, l'analyseur

syntaxique [Lê Ngoc et al., 2019] essaiera toutes les combinaisons possibles jusqu'à trouver un arbre complet.

Les arbres syntaxiques de Steedman décrivent les opérations d'application, présentées comme des inférences logiques, pour obtenir le symbole de phrase *S* à partir des catégories affectées aux mots :

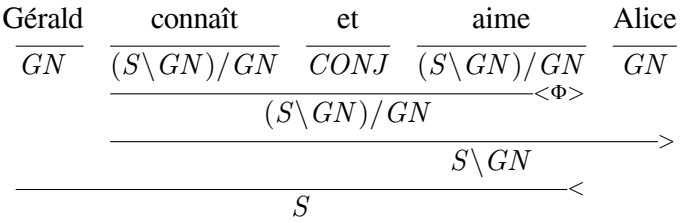


où les symboles > et < en marge des filets sont les règles d'inférence d'application à droite et à gauche. Si on complique un peu le verbe, la méthode est la même :



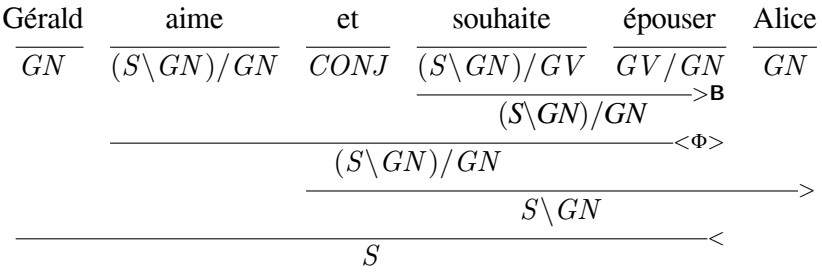
c'est-à-dire que le GV/GN d'« épouser » s'applique au *GN* d'« Alice », et on obtient ainsi un *GV* auquel s'applique le $(S\backslash GN)/GV$ de « souhaite ».

Le problème de la position relative des mots étant résolu par l'utilisation d'opérateurs d'application à gauche et à droite, il reste une contrainte : pour qu'une application puisse avoir lieu, il faut que les catégories concernées soient adjacentes dans le graphe. Ce qui n'est pas *a priori* le cas dans la phrase « Gérald connaît et aime Alice ». Alors Steedman introduit une nouvelle règle d'inférence pour la conjonction (Φ) qui a l'avantage d'agir en parfaite symétrie :

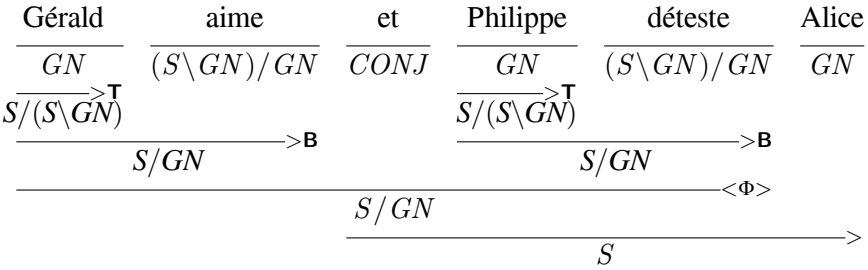


Mais que se passe-t-il lorsqu'on combine les deux dernières phrases pour obtenir « Gérald aime et souhaite épouser Alice » ? Les choses se compliquent parce que (a) d'une part on ne peut pas directement appliquer Φ puisque des deux côtés du « et » on n'a pas la même catégorie et (b) on ne peut pas non plus appliquer « épouser » à « Alice » puisque c'est la conjonction des deux verbes qui doit s'appliquer sur ce dernier, chose impossible s'il n'est plus là.

Il faut donc combiner le $(S\backslash GN)/GV$ de « souhaite » avec le GV/GN d'« épouser » pour obtenir un $(S\backslash GN)/GN$. Cela n'est rien d'autre qu'une *composition* et le combinateur dédié est le **B** (oiseau bleu). On obtient donc :



Un dernier exemple montre l'utilité d'un combinateur de *montée de type*, le **T** (tarin) :



Cette série d'exemples illustre à merveille l'utilisation des combinateurs pour former des règles d'inférence qui participent à l'analyse syntaxique de phrases complexes, de manière bien plus intuitive que l'approche des grammaires génératives.

4.3.7 Épilogue

Arrivé jusque là, le lecteur courageux de l'article mérite récompense, et celle-ci lui est donnée par deux curiosités :

1. la découverte par Barker [2001] d'une base de l'ensemble des combinateurs ne possédant qu'*un seul élément*. Ce combinateur, appelé **ι** (iota) est défini tout simplement par $\mathbf{\iota}x := x\mathbf{SK}$.
   Ce fait montre qu'il ne faut pas confondre *base de l'ensemble des combinateurs* et *base d'espace vectoriel*, où le nombre d'éléments indépendants est immuable et égal à la dimension de l'espace ;
2. un « secret pour les initiés » : il existe une communauté de personnes qui s'intéresse à un type particulier de langages de programmation, appelés *langages ésotériques*. Parmi ceux-ci, le langage de programmation Unlambda [Madore, 1990] qui est basé sur les combinateurs **S**, **K** et **I**. Le lecteur découvre que le mini-programme Unlambda

   ```
   ```s``s``sii`ki`k.*``s``s`ks``s`k`s`ks``s``s`ks``s`k`s`kr``s`k`sikk`k``s
   `ksk
   ```

   fournit des lignes d'astérisques dont les longueurs sont égales aux termes de la suite de nombres de Fibonacci.

C'est ainsi, en pleine zone magique, que se termine cet article riche en défis et en récompenses.

4.4 Perspectives

Nous voyons deux axes de perspectives concernant la vulgarisation.
   ```

En premier lieu la rédaction de nouveaux articles de vulgarisation. Les projets à court/moyen terme ne manquent pas : (a) la mathématisation des structures de la parenté, allant des groupes de Klein utilisés par Weil [1967] dans les *Structures élémentaires de la parenté* de Lévi-Strauss, aux langages formels de Liu [1986] et jusqu'aux au langage visuel des « figures » de Héran [2009] ; (b) la diagonale de Cantor avec son application dans la non-dénombrabilité des nombres réels, mais aussi, et surtout, dans le théorème d'incomplétude de Gödel ; (c) une présentation du langage de mathématiques formelles Mizar [Grabowski et al., 2010] avec quelques preuves formelles de théorèmes simples. Dans les trois cas il s'agit de sujets qui ne sont traditionnellement pas traités dans les vulgarisations, ou alors le sont très superficiellement, à cause de leur technicité, et c'est là tout le défi que nous souhaitons relever.

Mais cette production d'articles serait vaine sans une réflexion sur la vulgarisation en général. Plusieurs questions se posent :

- Pourquoi les articles de la revue *Tangente* font-ils irrémédiablement penser à de la presse à sensation et génèrent (du moins selon notre expérience personnelle) de la frustration chez le lecteur ?
- Dans un continuum qui va de la simple évocation approximative d'un résultat mathématique à sa démonstration complète et parfaitement rigoureuse, comment trouver le juste milieu qui ne frustre ni par sa platitude ni par sa difficulté insurmontable ?
- Pouvons-nous poser ce problème en tant que problème de représentation des connaissances ?
- Notre vision des mathématiques étant que le plus important dans un théorème et dans sa démonstration (les deux vont toujours ensemble), ce sont les « idées ». Comment modéliser cette notion pour produire un texte avec exactement le minimum de technicité requis pour pouvoir les transmettre au lecteur ?
- Enfin, quelle est la part de l'esthétique (voire de l'hédonique) dans l'enseignement (et la vulgarisation) « à la française » des mathématiques ? L'audace du bourbakisme produit-elle de la motivation et de l'ambition chez le lecteur, ou met-elle plutôt un frein à celles-ci ?

Bibliographie

Abeillé, A., Clément, L., et Toussenel, F. (2003). Building a treebank for French. In *Treebanks*, pages 165–187, Dordrecht. Kluwer.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., et Soroa, A. (2009). A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Alexander, S. (2013). The first-order syntax of variadic fuctions. *Notre Dame Journal of Formal Logic*, 54: 47–59.

André, J. et Borghi, B. (1990). Dynamic Fonts. *PostScript Language Journal*, 2(3): 4–6.

Anis, J. (1988a). *L'écriture, théorie et descriptions*. De Boeck, Bruxelles.

Anis, J. (1988b). *Texte et ordinateur*. De Boeck Université, Bruxelles.

Atanasiu, V. (2003). *Le phénomène calligraphique à l'époque du sultanat mamluk*. PhD thesis, École pratique des Hautes Études.

Ballier, N., Pacquetet, E., et Arnold, T. (2019). Investigating Keylogs as Time-Stamped Graphemics. In Haralambous, Y., editor, *Proceedings of Graphemics in the 21st Century, Brest 2018*, pages 353–365, Brest. Fluxus Editions.

Barker, C. (2001). Iota and jot : the simplest languages ? *The Esoteric Programming Languages Webring*. `https://perma.cc/M6US-33MA`.

Bayar, A. et Sami, K. (2010). Towards a Dynamic Font Respecting the Arabic Calligraphy. In Al Ajeeli, A. T. et Al-Bastaki, Y. A. L., editors, *Handbook of Research on E-services in the Public Sector : E-government Strategies and Advancements*, pages 359–379, Hershey PA. IGI Global.

Bélanger, V. et Gauvin, I. (2017). Comment et pourquoi utiliser l'arbre syntaxique en enseignement de la grammaire ? `https://perma.cc/NDC7-55R7`.

Bellamy-Royds, A. et al. (2018). Scalable Vector Graphics (SVG) 2. `https://www.w3.org/TR/SVG2/`.

Bigi, B. (2015). SPPAS – Multi-lingual Approaches to the Automatic Annotation of Speech. *The Phonetician – International Society of Phonetic Sciences*, 111–112: 54–69.

Bishop, T. et Cook, R. (2007). Wenlin CDL : Character Description Language. *Multilingual*, 18: 62–68.

Blanchard, G. (1980). *Pour une sémiologie de la typographie*. PhD thesis, École des Hautes Études en Sciences Sociales.

Boersma, P. et Weenink, D. (2001). PRAAT, a system for doing phonetics by computer. *Glot International*, 5(9–10): 341–347.

Bonfante, G., Guillaume, B., et Perrier, G. (2018). *Application of Graph Rewriting to Natural Language Processing*, volume 1 of *Logic, Linguistics and Computer Science Set*. Wiley.

Borleffs, E., Maassen, B. A., Lyytinen, H., et Zwarts, F. (2017). Measuring orthographic transparency and morphological-syllabic complexity in alphabetic orthographies : a narrative review. *Reading and writing*, 30: 1617–1638.

Borsley, R. D. et Börjars, K., editors (2011). *Non-Transformational Syntex. Formal and Explicit Models of Grammar*. Wiley-Blackwell.

Bourka, A., Drogkaris, P., et Agrafiotis, I., editors (2019). *Techniques et meilleures pratiques de pseydonymisation*. European Union Agency for Cybersecurity. `https://www.enisa.europa.eu/publications/pseudonymisation-techniques-and-best-practices_fr`.

Brosset, D., Claramunt, C., et Saux, É. (2008). Wayfinding in natural and urban environments : a comparative study. *Cartographica*, 43(1): 21–30.

Brousseau, G. (1998). *Théorie des situations didactiques*. La pensée sauvage, Paris.

Brown, K. et Miller, J. (2013). *The Cambridge Dictionary of Linguistics*. Cambridge University Press, Cambridge.

Carlisle, D., Ion, P., et Miner, R. (2014). Mathematical Markup Language (MathML) Version 3.0. `https://www.w3.org/TR/MathML3/`.

Chang, C.-H., Li, S.-Y., Lin, S., Huang, C.-Y., et Chen, J.-M. (2010). **以最佳化及機率分佈判斷漢字聲符之研究** (Automatic identification of phonetic complements for Chinese characters based on optimization and probability distribution). In *Proceedings of the 22nd Conference on Computational Linguistics and Speech Processing (ROCLING 2010), Puli, Nantou, Taiwan*, pages 199–209.

Chein, M. et Mugnier, M.-L. (2009). *Graph-based Knowledge Representation. Computational Foundations of Conceptual Graphs*. Advanced Information and Knowledge Processing Series. Springer.

Chepaitis, A., Griffiths, F., Wyatt, H., et O'Connell, W. (2004). Evaluation of tactile fonts for use by a visually impaired elderly population. 6: 111–134.

Chomsky, N. et Halle, M. (1968). *The Sound Pattern of English*. Harper & Row, New York.

Chou, S.-C., Gao, X.-S., et Zhang, J.-Z. (1996). Automated generation of readable proofs with geometric invariants, I. Multiple and shortest proof generation. *Journal of Automated Reasoning*, 17: 325–347.

Chou, S.-C., Gao, X.-S., et Zhang, J.-Z. (2000). A deductive database approach to automated geometry theorem proving and discovering. *Journal of Automated Reasoning*, 25: 219–246.

Chou, Y.-M., Hsieh, S.-K., et Huang, C.-R. (2007). Hanzi grid : toward a knowledge infrastructure for Chinese character-based cultures. In *Proceedings of the 1st international conference on Intercultural collaboration IWIC'07, Kyoto, Japan*, pages 133–145. Springer.

Chu, B.-F. (2003). **漢字基因 朱邦復 漢字基因工程** (Genetic engineering of Chinese characters). `http://cbflabs.com/down/show.php?id=26`.

Daille, B., Gaussier, É., et Lancé, J.-M. (1994). Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th International Conference on Computational Linguistics, Kyoto, Japan*, pages 515–521.

Davis, M. (2019a). Unicode Standard Annex 29. Unicode Text Segmentation. `https://www.unicode.org/reports/tr29/`.

Davis, M. (2019b). Unicode Standard Annex 9. Unicode Bidirectional Algorithm. `https://www.unicode.org/reports/tr9/`.

Deerwester, S., Dumais, S., et Furnas, G. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41: 391–407.

Dichy, J. (2019). On the Writing System of Arabic : The Semiographic Principle as Reflected in Nasẖī Letter Shapes. In YH, editor, *Proceedings of Graphemics in the 21st Century, Brest 2018*, pages 257–268, Brest. Fluxus Editions.

Duden (2018). Rechtschreiberegeln – Worttrennung. `https://perma.cc/6ZSW-2Z3Z`.

Dukes, K., Atwell, E., et Habash, N. (2013). Supervised Collaboration for Syntactic Annotation of Quranic Arabic. *Language Resources and Evaluation*, 47(1): 33–62.

Dürst, M. J. (1993). Coordinate-independent font description using Kanji as an example. *Electronic Publishing*, 6(3): 133–143.

Dürscheid, C. (2016). *Einführung in die Schriftlinguistik*. Vandenhoeck & Ruprecht, Göttingen.

Dürscheid, C. (2018). Bild, Schrift, Unicode. In Mensching, G., Lalande, J.-Y., Hermes, J., et Neuefeind, C., editors, *Sprache – Mensch – Maschine. Beiträge zu Sprache und Sprachwissenschaft, Computerlinguistik und Informationstechnologie für Jürgen Rolshoven aus Anlass seines sechsundsechzigsten Geburtstages*, pages 269–285, Köln. Kölner UniversitätsPublikationsServer.

Dürst, M. et Freytag, A. (2000). Unicode in XML and Other Markup Languages. `https://www.w3.org/TR/2000/NOTE-unicode-xml-20001215/`.

Dürst, M. et Suignard, M. (2005). Internationalized Resource Identifiers (IRIs). Request for Comments 3987.

Ferrucci, F. et al. (1998). Relation grammars : A formalism for syntactic and semantic analysis of visual languages. In Marriott, K. et Meyer, B., editors, *Visual language theory*, pages 219–243. Springer.

Ferrucci, F., Tortora, G., Tucci, M., et Vitiello, G. (1996). Symbol-Relation grammars : A formalism for graphical languages. *Information and Computation*, 131: 1–46.

Feys, R. (1946). La technique de la logique combinatoire. *Revue philosophique de Louvain*, 44: 74–103. `http://www.persee.fr/doc/phlou_0035-3841_1946_num_44_1_4039`.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., et Ruppin, E. (2002). Placing Search in Context : The Concept Revisited. *ACM Transactions on Information Systems*, 20: 116–131.

Frantzi, K. T., Ananiadou, S., et Tsujii, J. (1998). The C-value/NC-value Method of Automatic Recognition for Multi-word Terms. In *Proceedings of ECDL*, volume 1513 of *Springer Lecture Notes in Computer Science*, pages 585–604.

Fuhrman, O. et Boroditsky, L. (2010). Cross-cultural differences in mental representations of time : Evidence from an implicit nonlinguistic task. *Cognitive Science*, 34(8): 1430–1451.

Fujimura, O. et Kagaya, R. (1969). Structural patterns of Chinese characters. In *Proceedings of the International Conference on Computational Linguistics, Sånga-Säby, Sweden*, pages 131–148.

Fujiwara, Y., Suzuki, Y., et Morioka, T. (2004). Network of words. *Artificial Life and Robotics*, 7: 160–163.

Fusar-Poli, P. et al. (2013). The psychosis high-risk state : a comprehensive state-of-the-art review. *JAMA Psychiatry*, 70: 107–120.

Gabow, H. N., Galil, Z., Spencer, T., et Tarjan, R. E. (1986). Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6: 109–122.

Gabrilovich, E. et Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical Intelligence*.

Gao, Z. et al. (2008). Chinese WordNet. `http://www.aturstudio.com/wordnet/windex.php`.

Glück, H. (1987). *Schrift und Schriftlichkeit*. J.B. Metzler, Stuttgart.

Grabowski, A., Korniłowicz, A., et Naumowicz, A. (2010). Mizar in a nutshell. *Journal of Formalized Reasoning*, 3: 153–245.

Grzybek, P. et Rusko, M. (2009). Letter, Grapheme and (Allo-)Phone Frequencies : The Case of Slovak. *Glottotheory*, 2: 30–48.

Guillaume, B. et Perrier, G. (2015). Dependency parsing with graph rewriting. In *Proceedings of the 14th International Conference on Parsing Technologies, Bilbao, Spain*, pages 30–39.

Günther, H. (1988). *Schriftliche Sprache : Strukturen geschriebener Wörter und ihre Verarbeitung*. Niemeyer, Tübingen.

Halmos, P. R. (1970). How to write mathematics. *L'enseignement mathématique*, 16: 123–152.

Hellwig, O. (2010–2019). DCS—The Digital Corpus of Sanskrit. `http://www.sanskrit-linguistics.org/dcs/`.

Heninger, A. (2019). Unicode Standard Annex 14. Unicode Line Breaking Algorithm. `https://www.unicode.org/reports/tr14/`.

Héran, F. (2009). *Les figures de la parenté*. puf, Paris.

Honnibal, M. et Montani, I. (2017). spaCy 2 : Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hsieh, S.-K. (2006). *Hanzi, Concept and Computation : A Preliminary Survey of Chinese Characters as a Knowledge Resource in NLP*. PhD thesis, Universität Tübingen.

Huang, C.-R. (2003). Sinica BOW : Integrating bilingual WordNet and SUMO ontology. In *International Conference on Natural Language Processing and Knowledge Engineering*, pages 825–826.

Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., et Kanzaki, K. (2008). Development of the Japanese WordNet. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Iverach, L. et Rapee, R. M. (2014). Social anxiety disorder and stuttering : current status and future directions. *J. Fluency Disord.*, 40: 69–82.

Jones, K. S. (1981). *Information Retrieval Experiment*. Butterworths. `http://sigir.org/files/museum/Information_Retrieval_Experiment/pdfs/frontmatter.pdf`.

Kaplan, R. M. et Kay, M. (1994). Regular Models of Phonological Rule Systems. *Computational Linguistics*, 29: 331–378.

Knuth, D. E. (1984). *The TEXbook*. Addison-Wesley, Reading, MA.

Korver, N., Nieman, D. H., Becker, H. E., van de Fliert, J. R., Dingemans, P. H., de Haan, L., Spiering, M., Schmitz, N., et Linszen, D. H. (2010). Symptomatology and neuropsychological functioning in cannabis using subjects at ultra–high risk for developing psychosis and healthy controls. *Australian & New Zealand Journal of Psychiatry*, 44(3): 230–236.

Lacheret-Dujour, A., Kahane, S., et Pietrandrea, P. (2019). *Rhapsodie. A prosodic and syntactic treebank for spoken French*. John Benjamins, Amsterdam.

Le Yaouanc, J.-M., Saux, É., et Claramunt, C. (2010). A semantic and language-based representation of an environmental scene. *Geoinformatica*, 14(3): 333–352.

Leonard, H. S. et Goodman, H. N. (1937). A calculus of individuals (abstract). *The Journal of Symbolic Logic*, 2: 63–64.

Leonard, H. S. et Goodman, H. N. (1940). The calculus of individuals and its uses. *The Journal of Symbolic Logic*, 5: 45–55.

Lévi-Strauss, C. (1967). *Les structures élémentaires de la parenté*. Mouton & Co., Paris.

L'Homme, M.-C. (2004). *La terminologie : principes et techniques*. Presses de l'université de Montréal.

Li, J. et Zhou, J. (2007). Chinese character structure analysis based on complex networks. *Physica A : Statistical Mechanics and its Applications*, 380: 629–638.

Liu, H. (2008). Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9: 159–191.

Liu, P.-H. (1986). *Foundations of kinship mathematics*, volume 28 of *Monographs of the Institute of Ethnology*. Academia Sinica, Nakang, Taipei, Taiwan.

Lotman, J. (1977). *The Structure of the Artistic Text*, volume 7 of *Michigan Slavic Contributions*. The University of Michigan, Ann Arbor.

Lunde, K. (2008). *CJKV Information Processing*. O'Reilly, 2nd edition.

Mackenzie, C. E. (1980). *Coded Character Sets, History and Development*. Addison-Wesley, Reading, MA.

Madore, D. (1990). The Unlambda programming language. `http://www.madore.org/~david/programs/unlambda/`.

Manohar, K. et Thottingal, S. (2019). Malayalam Orthographic Reforms. Impact on Language and Popular Culture. In YH, editor, *Proceedings of Graphemics in the 21st Century, Brest 2018*, pages 329–351, Brest. Fluxus Editions.

Marneffe, M.-C. d., Connor, M., Silveira, N., Bowman, S. R., Dozat, T., et Manning, C. D. (2013). More Constructions, More Genres : Extending Stanford Dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 187–196, Prague. Matfyzpress.

Marriott, K. et Meyer, B. (1998). *Visual Language Theory*. Springer.

Martinet, A. (1970). La double articulation du langage. In *La linguistique synchronique*, pages 7–41, Paris. PUF.

Mathis, P. et Thierry, S. E. B. (2010). A formalization of geometric constraint systems and their decomposition. *Formal Aspects of Computing*, 22: 129–151.

Meletis, D. (2015). *Graphetik. Form und Materialität von Schrift*. Verlag Werner Hülsbusch, Glückstadt.

Meletis, D. (2019). *Naturalness in scripts and writing systems : Outlining a Natural Grapholinguistics*. PhD thesis, University of Graz.

Menanteau, B., editor (2011). *Édition des Instructions nautiques (procédure spécifique)*. SHOM.

Menanteau, B., editor (2013). *Définition des Instructions nautiques du SHOM (norme)*. SHOM.

Michon, P.-E. et Denis, M. (2001). When and why are visual landmarks used in giving directions ? In Montello, D. R., editor, *Spatial Information Theory*, volume 2205 of *Lecture Notes in Computer Science*, pages 292–305. Springer.

Moro, S. (2003). Surface or essence : Beyond the coded character set model. *Proceedings of the Glyph and Typesetting Workshop, Kyoto, Japan*, pages 26–35.

Mousavi Jazayeri, S., Michelli, P. E., et Abulhab, S. D. (2017). *A Handbook of Early Arabic Kufic Script*. Blautopf Publishing, New York.

Myers, J. (2019). *The Grammar of Chinese Characters. Productive Knowledge of Formal Patterns in an Orthographic System*. Routledge, London, New York.

Nachson, I. et Hatta, T. (2001). Directional tendencies of Hebrew, Japanese, and English readers. *Perceptual and Motor Skills*, 93: 178–180.

Nas, G. L. (1988). The effect on reading speed of word divisions at the end of a line. In van der Veer, G. C. et Mulder, G., editors, *Human-computer interaction : Psychonomic aspects*, pages 125–143. Springer.

Needleman, S. B. et Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48: 443–453.

Nehrlich, T. (2012). Phänomenologie der Ligatur. Theorie und Praxis eines Schriftelements zwischen Letter und Lücke. In Giertier, M. et Köppel, R., editors, *Von Lettern und Lücken : zur Ordnung der Schrift im Bleisatz*, pages 13–38, Paderborn. Wilhelm Fink.

Osborne, T. (2019). *A Dependency Grammar of English*. John Benjamins, Amsterdam/Philadephia.

Peebles, D. G. (2007). *SCML : A Structural Representation for Chinese Characters*. PhD thesis, Dartmouth College. TR2007–592.

Peixoto, T. P. (2014). The graph-tool Python library. *figshare*. `http://figshare.com/articles/graph_tool/1164194`.

Pelay, N. (2011). *Jeu et apprentissages mathématiques : Élaboration du concept de contrat didactique et ludique en contexte d'animation scientifique*. PhD thesis, Université Lyon 1.

Pelay, N. et Artigue, M. (2016). Quelle modélisation didactique de la vulgarisation des mathématiques ? In Barrier, T. et Chambris, C., editors, *Actes du séminaire national de didactique mathématique*, pages 228–242.

Pelay, N. et Boissière, A. (2015). Vulgarisation et enseignement des mathématiques dans le jeu Dobble. In Theis, L., editor, *Pluralités culturelles et universalité des mathématiques : enjeux et perspectives pour leur enseignement et leur apprentissage. Actes du colloque EMF2015*, pages 944–956.

Pelay, N. et Mercat, C. (2012). Quelle modélisation didactique de la vulgarisation des mathématiques ? In Dorier, J.-L. et Coutat, S., editors, *Enseignement des mathématiques et contrat social : enjeux et défis pour le* XXI*e siècle. Actes du colloque EMF2012*, pages 1914–1925.

Pollard, C. et Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., et Manning, C. D. (2020). Stanza : A Python Natural Language Processing Toolkit for Many Human Languages. `arXiv:2003.07082`.

Qin, L., Tong, C. S., Yin, L., et Ling, L. N. (2002). Decomposition for ISO/IEC 10646 ideographic characters. In *COLING'02: Proceedings of the 3rd workshop on Asian language resources and international standardization*. Association for Computational Linguistics.

Quaresma, P. (2011). Thousands of Geometric problems for geometric Theorem Provers (TGTP). In Schreck, P., Narboux, J., et Richter-Gebert, J., editors, *Automated Deduction in Geometry*, volume 6877 of *Lecture Notes in Computer Science*, pages 169–181. Springer.

Randell, D. A., Cui, Z., et Cohn, A. G. (1992). A spatial logic based on regions and connection. In *2rd Int. Conf. on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann.

Rezec, O. (2009). *Zur Struktur des deutschen Schriftsystems. Warum das Graphem nicht drei Funktionen gleichzeitig haben kann, warum ein <a> kein <ɑ> ist und andere Konstruktionsfehler des etablierten Beschreibungsmodells. Ein Verbesserungsvorschlag*. PhD thesis, Ludwig-Maximilians-Universität Munich.

Rittaud, B. (2015). Pour une « vulgaristique » des mathématiques. In Theis, L., editor, *Pluralités culturelles et universalité des mathématiques : enjeux et perspectives pour leur enseignement et leur apprentissage. Actes du colloque EMF2015*, pages 957–962.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System −Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall.

Rocha, J. et Fujisawa, H. (1996). Substructure shape analysis for Kanji character recognition. In *Advances in Structural and Syntactical Pattern Recognition*, volume 1121 of *Lecture Notes in Computer Science*, pages 361–370. Springer.

Salton, G. (1964). Information storage and retrieval. Technical report, The Computation Laboratory of Harvard University. `http://sigir.org/files/museum/pub-07/frontmatter.pdf`.

Sawicki, M., Suignard, M., Ishikawa, M., Dürst, M., et Texin, T. (2001). Ruby Annotation. `https://www.w3.org/TR/2001/REC-ruby-20010531/`.

Schindelin, C. (2007). *Zur Phonetizität chinesischer Schriftzeichen in der Didaktik des Chinesischen als Fremdsprache*, volume 13 of *SinoLinguistica*. iudicium, München.

Schopp, J. F. (2008). In Gutenbergs Fußstapfen : Translatio typographica. Zum Verhältnis von Typografie und Translation. *Meta*, 53: 167–183.

Siavash, S. (2017). SepidKhan : An alternate for Persian/Arabic braille. `https://typedrawers.com/discussion/2415/sepidkhan-an-alternate-for-persian-arabic-braille`.

Smullyan, R. (1985). *To mock a mockingbird, and other logic puzzles including an amazing adventure in combinatory logic*. Knopf.

Sousa Do Nascimento, S. (1999). *L'animation scientifique : essai d'objectivation de la pratique des associations de culture scientifique et technique françaises*. PhD thesis, Université Paris 6.

Sproat, R. (2000). *A Computational Theory of Writing Systems*. Cambridge University Press, Cambridge.

Steedman, M. (2000). *The syntactic process*. The MIT Press, Cambridge, MA.

Taft, M. et Zhu, X. (1997). Submorphemic processing in reading Chinese. *Journal of Experimental Psychology : Learning, Memory and Cognition*, 23: 761–775.

Tamaoka, K. et Yamada, H. (2000). The effects of stroke order and radicals on the knowledge of Japanese Kanji orthography, phonology and semantics. *Psychologia*, 43: 199–210.

The Unicode Consortium (2020). *The Unicode Standard, Version 13.0.0*. The Unicode Consortium. `http://www.unicode.org/versions/Unicode13.0.0/`.

Tversky, B. (2003). Structures of mental spaces. How people think about space. *Environment and behavior*, 35(1): 66–80.

Urieli, A. et Tanguy, L. (2013). L'apport du faisceau dans l'analyse syntaxique en dépendances par transitions : études de cas avec l'analyseur Talismane. In *Actes de la 20e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2013), Sables d'Olonne*, pages 188–201.

Valmaggia, L. R., Day, F. L., Jones, C., Bissoli, S., Pugh, C., Hall, D., Bhattacharyya, S., Howes, O., Stone, J., Fusar-Poli, P., et et al. (2014). Cannabis use and transition to psychosis in people at ultra–high risk. *Psychological Medicine*, 44(12): 2503–2512.

van den Bosch, A., Content, A., Daelemans, W., et de Gelder, B. (1994). Analysing orthographic depth of different languages using data-oriented algorithms. In *Proceedings of the 2nd International Conference on Quantitative Linguistics, Moscow*, pages 26–31.

van Gompel, M. et Reynaert, M. (2013). FoLiA : A Practical XML Format for Linguistic Annotation—a Descriptive and Comparative Study. *Computational Linguistics in the Netherlands Journal*, 3: 63–81.

Wang, J. C.-S. (1983). *Toward a generative grammar of Chinese character structure and stroke order*. PhD thesis, University of Wisconsin-Madison.

Wehde, S. (2000). *Typographische Kultur*. Max Niemeyer, Tübingen.

Weil, A. (1967). Sur l'étude algébrique de certains types de lois de mariage (système Murngin). In Lévi-Strauss [1967], p. 257–265.

Weingarten, R., Nottbusch, G., et Will, U. (2004). Morphemes, Syllables, and Graphemes in Written Word Production. *Trends in linguistics studies and monographs*, 157: 529–572.

Willett, P., Barnard, J. M., et Downs, G. M. (1998). Chemical similarity searching. *J. Chem. Inf. Comput. Sci.*, 38: 983–996.

Williams, C. et Bever, T. (2010). Chinese character decoding : a semantic bias ? *Reading and Writing*, 23: 589–605.

Wmffre, I. (2008). *Breton Orthographies and Dialects : The Twentieth-Century Orthography War in Brittany*, volume 18 of *Contemporary Studies in Descriptive Linguistics*. Peter Lang, Bern.

Wong, S. K. M., Ziarko, W., et Wong, P. C. N. (1985). Generalized vector spaces model in information retrieval. In *Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 85)*, pages 18–25. ACM.

Yu, S., Liu, H., et Xu, C. (2011). Statistical properties of Chinese phonemic networks. *Physica A : Statistical Mechanics and its Applications*, 390: 1370–1380.

Zhang, X., Zhao, J., et LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28.

Zhou, L. et Liu, Q. (2002). A character-net based Chinese text segmentation method. In *SEMANET '02 Proceedings of the 2002 workshop on Building and using semantic networks*, pages 1–6. Association for Computational Linguistics.

Publications de l'auteur

Bella, G. et YH (2007). Fontes intelligentes, textèmes et typographie dynamique. *Document numérique*, 9(3/4): 167–216.

Cennamo, I. et YH (2013). Enseigner la traduction humaine en s'inspirant de la traduction automatique. In *Tralogy II 2013: anticiper les technologies pour la traduction, Paris*.

Emani, C. et YH (2019). Deontic reasoning for legal ontologies. In Hitzler, P. et al., editors, *Proceedings of ESWC 2019 – The 18th European Semantic Web Conference, 2–6 June 2019, Portorož, Slovenia*, volume 11503 of *Springer Lecture Notes in Computer Science*, pages 209–224.

Fahed, L., Lenca, P., YH, et Lefort, R. (2020). Distant event prediction based on sequential rules. *Data Science and Pattern Recognition*, 4(1): 1–23.

Fahed, L., Lenca, P., YH, Lefort, R., et Tallec, M.-L. (2019). Prédiction d'évènements distants basée sur les règles séquentielles. In Rousset, M.-C. et Boudjeloud-Assala, L., editors, *Actes de la conférence EGC'2019*, pages 309–314. RNTI Éditions.

Haralambous, T. et YH (2003). Characters, glyphs and beyond. In *Glyph and Typesetting Workshop, Kyōto University 21st Century COE Program*.

Kacfah Emani, C. et YH (2017). Un système de questions-réponses dans le domaine légal : le cas des réglementations maritimes. *Traitement Automatique des Langues*, 58(2): 47–72.

Kluyev, V. et YH (2011). A query expansion technique using the EWC semantic relatedness measure. *Informatica*, 35: 401–406.

Kluyev, V. et YH (2012a). Accurate query translation for Japanese-English cross-language information retrieval. In *PECCS 2012 : 2nd international conference on pervasive and embedded computing and communications systems, Rome*, pages 214–219.

Kluyev, V. et YH (2012b). Query translation for CLIR : EWC vs. Google Translate. In *ICIST 2012: IEEE International Conference on Information Science and Technology, Wuhei, China*, pages 707–711.

Le Glaz, A., YH, Kim-Dufor, D.-H., Lenca, P., Billot, R., Taylor, R., Marsh, J., DeVylder, J., Walter, M., Berrouiguet, S., et Lemey, C. (2020). Machine learning and natural language processing in mental health : a systematic review. *Journal of Medical Internet Research*. doi :10.2196/15708.

Lê Ngoc, L. et YH (2019). CCG supertagging using morphological and dependency syntax information. In *CICLing 2019: 14th International Conference on Intelligent Text Processing and Computational Linguistics, 7–13 April 2019, La Rochelle*. (To appear).

Lê Ngoc, L., YH, et Lenca, P. (2019). Towards a DRS parsing framework for French. In Elnagar, A. et Abdel-Maguid, M., editors, *IEEE ANLP 2019 : The Third IEEE International Workshop on Advances in Natural Language Processing*. IEEE Conference Publishing Services.

Lyubareva, I., Rochelandet, F., et YH (2020). Une étude exploratoire sur l'information d'actualité. *Revue d'Economie Industrielle*. en révision.

Mehta, A., Bella, G., et YH (2003). Adapting Omega to OpenType fonts. *TUGboat*, 24(3): 550–556.

Plaice, J., Swoboda, P., YH, et Bella, G. (2004). Moving Ω to an object-oriented platform. *Springer Lecture Notes in Computer Science*, 3130: 17–26.

Plaice, J., Swoboda, P., YH, et Rowley, C. (2003a). A multidimensional approach to typesetting. *TUGboat*, 24(1): 105–114.

Plaice, J. et YH (1996). The latest developments in Ω. *TUGboat*, 17(2): 181–183.

Plaice, J. et YH (1997). Methods for processing languages with Ω. In *Proceedings of International Symposium on Multilingual Information Processing '97*, pages 115–128, Tsukuba (Japan).

Plaice, J. et YH (2003). Generating multiple outputs from Ω. *TUGboat*, 24(3): 512–518.

Plaice, J., YH, et Rowley, C. (2003b). An extensible approach to high-quality multilingual typesetting. In *IEEE Research Issues in Data Engineering : Multi-lingual Information Management, Hyderabad*.

Quaresma, P. et YH (2012). Geometry construction recognition by the use of semantic graphs. In *RECPAD 2012: 18th Portuguese Conference on Pattern Recognition, Coimbra*, pages 47–48.

Sauvage-Vincent, J., YH, et Puentes, J. (2015). Sentence ordering in electronic navigational chart companion text generation. In *ENLG 2015: 15th European Workshop on Natural Language Generation, Stroudsburg*, pages 66–70.

YH (1989). TEX and Latin alphabet languages. *TUGboat*, 10(3): 342–345.

YH (1990a). Arabic, Persian and Ottoman TEX. *TUGboat*, 11(4): 520–524.

YH (1990b). *Coformalité modérée et formalité des CW-complexes de dimension finie*. PhD thesis, Université de Lille 1.

YH (1990c). Coformalité modérée et formalité des CW-complexes de dimension finie. *C.R.A.S.*, 311: 365–368.

YH (1991a). TEX and those other languages…. *TUGboat*, 12(4): 539–548.

YH (1991b). On TEX and Greek…. *TUGboat*, 12(2): 224–226.

YH (1991c). Perturbation d'algèbres de Lie différentielles et coformalité modérée. *Bulletin de la Société mathématique belge*, 43: 59–67.

YH (1991d). ScholarTEX. *Cahiers GUTenberg*, 10–11: 69–70.

YH (1991e). Typesetting Old German : Fraktur, Schwabacher, Gotisch and Initials. *TUGboat*, 12(1): 129–138.

YH (1992a). TEX conventions concerning languages. *TEX and TUG News*, 1(4): 3–10.

YH (1992b). Hyphenation patterns for ancient Greek and Latin. *TUGboat*, 13(4): 457–469.

YH (1992c). Les langues « exotiques ». In *École d'été Didot. Dessin de caractères assisté par ordinateur. Rennes 6–10 juillet 1992*.

YH (1992d). Towards the revival of traditional Arabic typography. In *Proceedings of the 7th European TEX Conference*, pages 293–305, Prague (Czech Republic).

YH (1992e). Typesetting the Holy Qur'ān with TEX. In *Proceedings of the 3rd International Conference and Exhibition on Multilingual computing (Arabic and Roman script)*, pages 110–125, Durham (United Kingdom).

YH (1992f). A typewriter font for the Macintosh 8-bit font table. *TUGboat*, 13(4): 476–477.

YH (1992g). Virtual fonts not for grand wizards only. *MAPS Nederlandstalige TEX Gebruikersgroep*, 93(1): 114–119.

YH (1993a). TEX on the Macintosh. *Notices of the American Mathematical Society*, 40(10): 1353–1360.

YH (1993b). Conventions concernant les polices DC et les langages naturels. *Cahiers GUTenberg*, 15: 53–61.

YH (1993c). The Khmer script tamed by the lion (of TEX). *TUGboat*, 14(3): 260–270.

YH (1993d). Parametrization of PostScript fonts through METAFONT, an alternative to Adobe Multiple Master. *Electronic Publishing—Origination, Dissemination, and Design*, 6(3): 145–147.

YH (1994a). Indica, an Indic preprocessor for TEX—Sinhalese TEX. *TUGboat*, 15(4): 447–458.

YH (1994b). Tiqwah, a typesetting system for biblical Hebrew, based on TEX. In *Actes du Quatrième Colloque International Bible et informatique, matériel et matière*, pages 445–470, Amsterdam.

YH (1994c). The traditional Arabic typecase extended to the Unicode set of glyphs. *Electronic Publishing—Origination, Dissemination, and Design*, 8(2/3): 125–138.

YH (1994d). Typesetting biblical Hebrew with TEX. *TUGboat*, 15(3): 174–191.

YH (1994e). Typesetting Khmer. *Electronic Publishing—Origination, Dissemination, and Design*, 7(4): 197–215.

YH (1994f). Un système TEX berbère. *Études et documents berbères*, 11: 43–54.

YH (1994g). Was ist ScholarTEX ? In *Offizin, Schriftenreihe zu TEX, LATEX und METAFONT*, pages 31–47, München. Addison-Wesley.

YH (1995a). Sabra, a Syriac TEX system. In *Proceedings of SyrCOM-95, First International Forum on Syriac Computing*, pages 3–24, Washington.

YH (1995b). Some METAFONT techniques. *TUGboat*, 16(1): 46–53.

YH (1995c). Tour du monde des ligatures. *Cahiers GUTenberg*, 22: 69–70.

YH (1996). Complexes consonantiques cambodgiens du dictionnaire cambodgien-français d'Alain Daniel. In *Proceedings of International Symposium on Multilingual Information Processing '96*, pages 249–258, Tsukuba (Japan).

YH (1997). The traditional Arabic typecase, Unicode, TEX and METAFONT. *TUGboat*, 18(1): 17–29.

YH (1998). Simplification of the Arabic script : Two different approaches and their implementations. In *Electronic Publishing, Artistic Imaging, and Digital Typography*, volume 1375 of *Springer Lecture Notes in Computer Science*, pages 138–156.

YH (1999a). From Unicode to typography, a case study : the Greek script. In *Proceedings of International Unicode Conference XIV, Boston*, pages b.10.1–b.10.36. Russian translation : "От Юникода к типографике : исследование греческой письменности," *Feature* 9:5–29, 2012 `https://issuu.com/typefamily/docs/feature_calt`.

YH (1999b). Une police mathématique pour la Société mathématique de France : le SMF Baskerville. *Cahiers GUTenberg*, 32: 5–10.

YH (1999c). Ὁ σύγχρονος τυπογράφος. *ΗΥΦΕΝ*, 2: 1–25.

YH (2000a). Druckfatz in gebrochenen Schriften. In *Proceedings of TEX-Tagung DANTE 2000, Clausthal*, pages 24–37. `https://perma.cc/66WM-2SWS`.

YH (2000b). Unicode, XML, TEI, Ω. In *Proceedings of the International Unicode Conference XVI, Amsterdam*, pages b.7.1–b.7.23.

YH (2002a). Foreword. In Syropoulos, A., Tsolomitis, A., et Sofroniou, N., editors, *Digital Typography Using LATEX*, pages xiv–xxiii. Springer.

YH (2002b). Guidelines and suggested amendments to the Greek Unicode tables. In *Proceedings of the International Unicode Conference XXI, Dublin*, pages 1–25.

YH (2002c). Keeping Greek typography alive. In *Proceedings of the First International Conference on Typography & Visual Communication, Thessaloniki*, pages 63–91. Ἐκδόσεις Πανεπιστημίου Μακεδονίας.

YH (2002d). Unicode et typographie : un amour impossible. *Document numérique*, 6(3/4): 107–139.

YH (2004a). *Fontes & codages. Glyphes et caractères à l'ère du numérique*. O'Reilly France, Paris.

YH (2004b). Voyage au centre de TEX : composition, paragraphage, césure. *Cahiers GUTenberg*, 44–45: 3–53.

YH (2005). (Dé)codages typographiques (Revue des Rencontres internationales de Lure). *TYP Observatoire typo_graphique*, 2: 103–117.

YH (2006a). Infrastructure for high-quality Arabic typesetting. *TUGboat*, 27(2): 167–175.

YH (2006b). New hyphenation techniques in Ω_2. *TUGboat*, 27(1): 98–103.

YH (2006c). Οἱ ἀναφορὲς τῶν μονοτονιστῶν στὸν Βιλαμόβιτς καὶ στὸν Γεώργιο Χατζιδάκι. *Εὐθύνη*. Paper accepted for publication by *Εὐθύνη* but never published `https://perma.cc/H8EQ-H2CD`.

YH (2007). *Fonts & Encodings. From Advanced Typography to Unicode and Everything in Between*. O'Reilly, Sebastopol, CA.

YH (2010). D'Unicode au Web sémantique : les dernières technologies du Web au service de la terminologie. In *GLAT 2010: le multiculturalisme et le rôle des langues spécialisées, Lisbonne*, pages 241–262.

YH (2011). Seeking meaning in a space made out of strokes, radicals, characters and compounds. In *9th International Symposium on Spatial Media, Aizu-Wakamatsu*, pages 1–13. `https://perma.cc/A6BY-NL4D`.

YH (2012). Text mining methods applied to mathematical texts. In *Conference on Intelligent Computer Mathematics, Bremen*.

YH (2013). New perspectives in sinographic language processing through the use of character structure. In *CICLing 2013: 14th International Conference on Intelligent Text Processing and Computational Linguistics, Samos*, volume 7816 of *Springer Lecture Notes in Computer Science*, pages 201–217.

YH (2014). A simple Arabic typesetting system for mixed Latin/Arabic documents. *TUGboat*, 35(3): 277–283.

YH (2015). Les mathématiques de la langue : l'approche formelle de Montague. *Quadrature*, 98: 9–19.

YH (2017). Gestion de la répétition dans les correspondances graphème-phonème et graphème-morphème. *Repères DORIF*, 13. `http://www.dorif.it/ezine/ezine_articles.php?art_id=346`.

YH (2018). TEX as a path, a talk given at Donald Knuth's 80th birthday celebration symposium. *TUGboat*, 39(1): 8–15.

YH (2019a). Approches et applications de la graphématique. In Waldeck, R., editor, *Méthodes et interdisciplinarité*, Méthodologies de modélisation en sciences sociales, pages 135–151. ISTE Éditions.

YH (2019b). Classification de textes anglais L2 par niveau de compétence langagière. In Roxin, I., Tajariol, F., Hosu, I., et Pélissier, N., editors, *Actes du colloque Information, communication et humanités numériques. Enjeux et défis pour un enrichissement épistémologique*, pages 129–142, Cluj-Napoca. Accent.

YH, editor (2019c). *Graphemics in the 21st Century 2018. Proceedings*, Brest. Fluxus Editions.

YH (2019d). Ne vous moquez pas de l'oiseau moqueur : un aperçu de la logique combinatoire, avec des applications en linguistique mathématique. *Quadrature*, 113: 22–34.

YH (2019e). Phonocentrism in Greece : Side effects of two centuries of diglossia. poster presented at *AWLL12*, Cambridge, UK, `https://hal.archives-ouvertes.fr/hal-02480230`.

YH (2019f). *Ὁδηγὸς γραφῆς γιὰ τὴν ἑλληνικὴ γλώσσα*. Ἐκδόσεις Ἄγρα. (To appear).

YH (2020). Grapholinguistics, TEX, and a June 2020 conference. *TUGboat*, 41(1): 12–19.

YH et Bella, G. (2005a). Injecting information into atomic units of text. In *ACM Symposium on Document Engineering, Bristol*.

YH et Bella, G. (2005b). Omega becomes a texteme processor. In *Proceedings of EuroTEX 2005 Conference, Pont-à-Mousson*.

YH, Bella, G., et Gulzar, A. (2006). Open-belly surgery in Ω_2. In *Proceedings of EuroTEX 2006 Conference, Debrecen*.

YH et Dichy, J. (2019). Graphemic methods for gender-neutral writing. In YH, editor, *Proceedings of Graphemics in the 21st Century, Brest 2018*, pages 41–89, Brest. Fluxus Editions.

YH et Dürst, M. (2019). Unicode from a linguistic point of view. In YH, editor, *Proceedings of Graphemics in the 21st Century, Brest 2018*, pages 167–183, Brest. Fluxus Editions.

YH, Elidrissi, Y., et Lenca, P. (2014a). Arabic language text classification using dependency syntax-based feature selection. In *CITALA 2014 : 5e Conférence Internationale sur le Traitement Automatique de la Langue Arabe, Oujda*, pages 31–40.

YH et Kluyev, V. (2011). A semantic relatedness measure based on combined encyclopedic, ontological and collocational knowledge. In *International Joint Conference on Natural Language Processing, Chiang Mai*.

YH et Kluyev, V. (2013). Thematically reinforced explicit semantic analysis. In *CICLing 2013: 14th International Conference on Intelligent Text Processing and Computational Linguistics, Samos*, volume 4 of *Computational Linguistics and Applications*, pages 79–94.

YH et Laugier, M. (1994). TEX innovations by the Louis-Jean printing house. *TUGboat*, 15(4): 438–443.

YH et Lavagnino, E. (2011). La réduction de termes complexes dans les langues de spécialité. *Traitement Automatique des Langues*, 52(1): 37–68.

YH, Lemey, C., Lenca, P., Billot, R., et Kim-Dufor, D.-H. (2020). Using dependency syntax-based methods for automatic detection of psychiatric comorbidities. In Kokkinakis, D., Lundholm Fors, K., Themistocleous, H., Antonsson, M., et Eckerström, M., editors, *Resources and Processing of linguistic, para-linguistic and extra-linguistic data from people with various forms of cognitive/psychiatric/developmental impairments*, pages 142–150. European Language Resources Association.

YH et Lenca, P. (2014). Text classification using association rules, dependency pruning and hyperonymization. In *DMNLP 2014: Workshop on Interactions between Data Mining and Natural Language Processing*, volume 1202 of *CEUR Workshop Proceedings*, pages 65–80.

YH et N'zi, E. (2019). Salience-induced term-driven serendipitous web exploration. In *CICLing 2019: 14th International Conference on Intelligent Text Processing and Computational Linguistics, 7–13 April 2019, La Rochelle*. (To appear).

YH et Plaice, J. (1994). First applications of Ω : Adobe Poetica, Arabic, Greek, Khmer. *TUGboat*, 15(3): 344–352.

YH et Plaice, J. (1995). Omega, une extension de TEX incluant Unicode et des filtres du type lex. *Cahiers GUTenberg*, 20: 55–79.

YH et Plaice, J. (1996). Ω Times and Ω Helvetica fonts under development, step one. *TUGboat*, 17(2): 126–146.

YH et Plaice, J. (1997). Multilingual typesetting with Ω, a case study : Arabic. In *Proceedings of International Symposium on Multilingual Information Processing '97*, pages 137–154, Tsukuba (Japan).

YH et Plaice, J. (1998). The design and use of a multiple-alphabet font with Ω. In *Electronic Publishing, Artistic Imaging, and Digital Typography*, volume 1375 of *Springer Lecture Notes in Computer Science*, pages 126–137.

YH et Plaice, J. (1999a). Produire du MathML et autres *ML à partir d'Ω : Ω se généralise. *Cahiers GUTenberg*, 33–34: 173–182.

YH et Plaice, J. (1999b). Δεκαοκτὼ γραμματοσειρὲς προτιμοῦν τὸ Ὠμέγα. *Εὔτυπον*, 2: 23–38.

YH et Plaice, J. (1999c). Ὁ ἄυλος αὐλός, ἢ τὸ Ὠμέγα καὶ τὰ ἑλληνικά. *Εὔτυπον*, 2: 1–27.

YH et Plaice, J. (2001a). Traitement automatique des langues et composition sous Ω. *Cahiers GUTenberg*, 39–40: 139–166.

YH et Plaice, J. (2001b). **製版・文書処理システム** Ω. *BIT*, 4: 137–152. Kyoritsu Shuppan, Tokyo.

YH et Plaice, J. (2002). Low-level Devanāgarī support for Ω—Adapting devnag. *TUGboat*, 23(1): 50–56.

YH et Plaice, J. (2003a). Omega and OpenType fonts. In *Glyph and Typesetting Workshop, Kyōto University 21st Century COE Program*.

YH et Plaice, J. (2003b). XLATEX, a DTD/schema which is very close to LATEX. *TUGboat*, 24(3): 369–376.

YH, Plaice, J., et Braams, J. (1995). Never again active characters ! Ω-Babel. *TUGboat*, 16(4): 418–427.

YH et Quaresma, P. (2014). Querying geometric figures using a controlled language, ontological graphs and dependency lattices. In *CICM 2014: Proceedings of the Conferences on Intelligent Computer Mathematics, Coimbra*, volume 8543 of *Springer Lecture Notes in Computer Science*, pages 298–311.

YH et Quaresma, P. (2018a). Geometric figure mining via conceptual graphs. preprint.

YH et Quaresma, P. (2018b). Geometric search in TGTP. In Li, H., editor, *Proceedings of the 12th International Conference on Automated Deduction in Geometry, 11–14 September 2018, Nanning, China*, pages 19–25.

YH et Rahtz, S. (1995). HTML → LaTeX → PDF, ou l'entrée de TeX dans l'ère de l'hypertexte. *Cahiers GUTenberg*, 19: 127–147. Dutch translation "HTML → LaTeX → PDF, of de intrede van TeX in het hypertext tijdperk," *MAPS* 14(1) :99-108, 1995. English translation : "LaTeX, HTML and PDF, or the entry of TeX into the world of hypertext," *TUGboat* 16(2) :162–173, 1995.

YH, Sauvage-Vincent, J., et Puentes, J. (2014b). INAUT, a controlled language for the French coast pilot books *Instructions nautiques*. In *CNL 2014: Proceedings of the 4th Workshop on Controlled Natural Language, Galway*, volume 8625 of *Springer Lecture Notes in Computer Science*, pages 102–111.

YH, Sauvage-Vincent, J., et Puentes, J. (2017). A hybrid (visual/natural) controlled language. *Language Resources and Evaluation*, 51(1): 93–129.

YH et Thull, K. (1989). Typesetting modern Greek with 128 character codes. *TUGboat*, 10(3): 354–359. Followed by "Typesetting modern Greek—An update," *TUGboat* 11(1): 26, 1989.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Université de Bretagne Occidentale

Titre : Des graphèmes à la langue et à la connaissance

Mots clés : grapholinguistique, représentation et gestion des connaissances, langages contrôlés visuels, traitement automatique du langage naturel, vulgarisation scientifique

Résumé : Ce document comporte quatre chapitres autour de l'écriture, de la langue, de la connaissance et de la vulgarisation des disciplines scientifiques qui s'intéressent à ces sujets. Le premier chapitre, intitulé «texte informatique», s'intéresse à une discipline émergente, la grapholinguistique, à ses applications informatiques à travers le codage de caractères Unicode, au cas particulier de l'écriture chinoise, et à une proposition d'extension d'Unicode, les textèmes.
Dans le deuxième chapitre il est question de langages contrôlés avec une composante visuelle et de représentation de connaissances dans deux cas d'étude : la modélisation des figures géométriques euclidiennes à travers les graphes conceptuels variadiques et la preuve de concept d'un langage destiné à l'interaction entre les navigants et la base de connaissances du Service hydrographique et océanographique de la marine.
Dans la suite, nous abordons le traitement automatique de la langue par le biais de trois cas d'étude : les plongements de mots dans des espaces wikipédiques catégorisés, l'exploration sérendipiteuse du Web et l'utilisation d'interstices syntaxiques pour automatiser l'évaluation de comorbidités dans des textes d'entretiens psychiatriques, en vue d'une détection précoce de la schizophrénie.
Enfin, le document se termine par un chapitre intitulé «Exercices de vulgarisation», où sont décrites, dans un cadre didactique, deux tentatives de vulgarisation de sujets scientifiques réputés difficiles (la sémantique formelle de Montague et la logique combinatoire) à destination d'un public de lycéens et d'élèves du 1er cycle universitaire.

Title : From graphemes to language and to knowledge

Keywords : grapholinguistics, knowledge representation and management, visual controlled languages, natural language processing, science popularization

Abstract : This document consists of four chapters around the topics of writing, language, knowledge and popularization of disciplines interested in these topics.
The first chapter, entitled "Computer text," deals with the emergent discipline of grapholinguistics, its applications in computing by means of the Unicode character encoding, with the special case of the Chinese writing system and with a Unicode extension proposal: textemes.
In the second chapter we focus on controlled natural languages with a visual component and on knowledge representation in two study cases: the modeling of Euclidean geometry figures through variadic conceptual graphs and the proof of concept of a language dedicated to the interaction between navigators and the knowledge base of the French Oceanographic and Hydrographic Service.
In the third chapter we deal with natural language processing through three study cases: word embeddings in Wikipedic categorized spaces, serendipitous exploration of the Web and the use of syntactic gaps to automate evaluation of comorbidities in psychiatric interview texts, in order to achieve early detection of schizophrenia.
Finally, the document ends with a chapter entitled "Popularization exercises" where we describe, in a didactic frame, two attempts to popularize notoriously difficult scientific topics (Montague formal semantics and combinatory logic) to high school and undergraduate students.