Définitions et caractérisations de modèles à base d'analogies pour l'apprentissage automatique des langues naturelles

Nicolas Stroppa

ÉCOLE NATIONALE SUPÉRIEURE DES TÉLÉCOMMUNICATIONS
DÉPARTEMENT INFORMATIQUE ET RÉSEAUX

Thèse présentée en vue de l'obtention du grade de
Docteur de l'École Nationale Supérieure des Télécommunications
(Spécialité Informatique)

Définitions et caractérisations de modèles à base d'analogies pour l'apprentissage automatique des langues naturelles

Nicolas STROPPA

soutenue publiquement le 4 novembre 2005 devant un jury composé de :

Jacques SAKAROVITCH	Président
François DENIS	Rapporteurs
Pierre ZWEIGENBAUM	
Laurent MICLET	Examinateurs
Vito PIRRELLI	
François YVON	Directeur de thèse

Remerciements

Nobody said it was easy,
No one ever said it would be this hard.

— The Scientist *(Coldplay)*

Au cours des trois dernières années, j'ai accumulé une quantité non négligeable de dettes.

En premier lieu, je pense à François Yvon, directeur de thèse disponible et à l'écoute ; ses nombreuses remarques constructives sont à l'origine de bien des questionnements et stimulations ayant contribué à l'accomplissement du présent travail.

Je tiens à exprimer ma profonde reconnaissance à Jacques Sakarovitch pour m'avoir fait l'honneur d'assurer la présidence du jury lors de la soutenance ; je le remercie également de m'avoir fait bénéficier à plusieurs reprises de son expertise sur la théorie des automates. J'adresse mes plus sincères remerciements à François Denis et Pierre Zweigenbaum qui, sans me connaître, ont accepté spontanément de rapporter ce travail ; j'essaierai de faire le meilleur usage de leurs commentaires extrêmement précis et précieux. Je remercie très chaleureusement Laurent Miclet, dont les marques d'encouragement n'ont jamais manqué, ainsi que Vito Pirrelli qui n'a pas hésité à m'accorder de son temps pour échanger librement avec moi.

Cette thèse a été préparée dans le département Informatique et Réseaux de l'ENST ; qu'il me soit permis d'en citer quelques membres. Tout d'abord mes camarades de bureau : Nadine, Romain l'omni-expert (encore merci pour ta patience face à mes sollicitations quotidiennes), Nicolas, Tuan-Anh, Baptiste et Bilel. Encore des doctorants : Jean-Philippe, Laleh, Cyril, Alexandre, Axel, Pascal, Lucille, Éric, Thomas et le binôme dareausien de TSI : Loïs et Thomas. Du côté des permanents : Jean-Louis (merci pour ta disponibilité et ta bonne humeur, je regretterai nos longues discussions « enflammées »), Alain, Georges, Talel, Jean-Marc, Irène, Olivier, Antoine, Jean, Hayette, Céline et Sophie.

Certaines personnes m'ont accordé leurs confiances en me donnant la chance d'enseigner ; je leur en suis très reconnaissant. À l'Université Paris I : Susana ; au département InfRes : Annie, Irène, Sylvie, François, Éric et Laurent.

Quelques imprudents ont fait l'erreur d'accepter de relire ce manuscrit : Loïs mériterait incontestablement une palme pour cette tâche (encore merci Loïs) ainsi que mon père qui y a consacré un temps considérable (merci beaucoup Papa) ; j'ai également bénéficié de remarques pertinentes de la part de Nicolas et Xavier.

Les logiciels libres simplifient énormément le travail d'un doctorant en informatique. Je tiens à exprimer ma reconnaissance à leurs nombreux concepteurs et développeurs ; je pense en particulier aux personnes impliquées d'une manière ou d'une autre dans la réalisation des logiciels ou systèmes suivants : TEX, LATEX, EMACS, GCC, KDE, TELLICO, FIREFOX, PYTHON, LINUX.

De nombreux amis m'ont entouré au cours de ces trois années. Pour éviter de faire réference à Jean Noubli, je préfère adresser un remerciement global dirigé vers les personnes présentes (en chair ou en esprit) à la soirée Vins&Fromage ; elles constituent un échantillon assez représentatif de mes créanciers.

Ma famille a toujours été présente à mes côtés : mes grands-parents, mes parents, mes deux frères Bastien et Florent ainsi que ma « parraine » Janine. Qu'ils sachent qu'ils sont pour beaucoup dans l'énergie qui m'anime.

Enfin, merci surtout à toi Gaëlle, toi qui m'a accompagné dans cette succession de jours, toi qui a supporté mes *fouthèses* si longtemps et à qui je peux enfin dire : « j'en ai fini de te parler d'*allégorithmes*, il est désormais temps pour moi de t'*alcâliner* [1]. »

1. Extraits du dictionnaire de mots-valises d'Alain Créhange, « *Le pornithorynque est un salopare* » (`http://perso.wanadoo.fr/alain.crehange/frmotsval.html`).
   - **Alcâliner.** Donner des preuves de tendresse pendant plus longtemps qu'un amant ordinaire.
   - **Allégorithme.** Représentation symbolique d'une idée au moyen d'une suite d'opérations mathématiques élémentaires.
   - **Fouthèse.** Doctrine qui n'a de valeur que dans l'esprit quelque peu dérangé de son auteur.

Résumé

Le panorama du Traitement Automatique des Langues est dominé par deux familles d'approches : dans la première, la connaissance linguistique s'exprime sous forme de règles (grammaticales pour le traitement syntaxique, d'inférence pour le traitement sémantique, etc.), et de représentations sur lesquelles ces règles opèrent. La deuxième repose sur l'hypothèse d'un modèle probabiliste sous-jacent aux données, modèle dont les paramètres s'infèrent à partir de corpus de données linguistiques annotées. Ces deux familles de méthodes, bien qu'efficaces pour nombre d'applications, présentent de sérieuses limitations. Pour la première, il s'agit de la difficulté et du coût de construction des bases de connaissances de haute qualité : les experts sont rares et la connaissance accumulée sur un domaine X ne se transporte pas toujours simplement sur un autre domaine Y. Les méthodes probabilistes, quant à elles, ne traitent pas naturellement les objets fortement structurés, ne prévoient pas d'inclusion de connaissances linguistiques explicites, et surtout, reposent lourdement sur le choix a priori d'un certain modèle, puisqu'utilisant principalement des techniques de statistiques paramétriques.

Dans le cadre d'un apprentissage automatique de données linguistiques, des modèles inférentiels alternatifs ont alors été proposés qui remettent en cause le principe d'abstraction opéré par les règles ou les modèles probabilistes. Selon cette conception, la connaissance linguistique reste implicitement représentée dans le corpus accumulé. Dans le domaine de l'Apprentissage Automatique, les méthodes suivant les même principes sont regroupées sous l'appellation d'apprentissage « paresseux ». Ces méthodes reposent généralement sur le biais d'apprentissage suivant : si un objet Y est « proche » d'un objet X, alors son analyse $f(Y)$ est un bon candidat pour $f(X)$. Alors que l'hypothèse invoquée se justifie pour les applications usuellement traitées en Apprentissage Automatique, la nature structurée et l'organisation paradigmatique des données linguistiques suggèrent une approche légèrement différente. Pour rendre compte de cette particularité, nous étudions un modèle reposant sur la notion de « proportion analogique ». Dans ce modèle, l'analyse $f(T)$ d'un nouvel objet T s'opère par identification d'une proportion analogique avec des objets X, Y et Z déjà connus. L'hypothèse analogique postule ainsi que si « X est à Y ce que Z est à T », alors « $f(X)$ est à $f(Y)$ ce que $f(Z)$ est à $f(T)$ ». Pour inférer $f(T)$ à partir des $f(X)$, $f(Y)$, $f(Z)$ déjà connus, on résout l'« équation

analogique » d'inconnue I : « $f(X)$ est à $f(Y)$ ce que $f(Z)$ est à I ».

Nous présentons, dans la première partie de ce travail, une étude de ce modèle de proportion analogique au regard d'un cadre plus général que nous qualifierons d'« apprentissage par analogie ». Ce cadre s'instancie dans un certain nombre de contextes : dans le domaine des sciences cognitives, il s'agit de raisonnement par analogie, faculté essentielle au cœur de nombreux processus cognitifs ; dans le cadre de la linguistique traditionnelle, il fournit un support à un certain nombre de mécanismes tels que la création analogique, l'opposition ou la commutation ; dans le contexte de l'apprentissage automatique, il correspond à l'ensemble des méthodes d'apprentissage paresseux. Cette mise en perspective offre un éclairage sur la nature du modèle et les mécanismes sous-jacents.

La deuxième partie de notre travail propose un cadre algébrique unifié, définissant la notion de proportion analogique. Partant d'un modèle de proportion analogique entre chaînes de symboles, éléments d'un monoïde libre, nous présentons une extension au cas plus général des semigroupes. Cette généralisation conduit directement à une définition valide pour tous les ensembles dérivant de la structure de semigroupe, permettant ainsi la modélisation des proportions analogiques entre représentations courantes d'entités linguistiques telles que chaînes de symboles, arbres, structures de traits et langages finis. Des algorithmes adaptés au traitement des proportions analogiques entre de tels objets structurés sont présentés. Nous proposons également quelques directions pour enrichir le modèle, et permettre ainsi son utilisation dans des cas plus complexes.

Le modèle inférentiel étudié, motivé par des besoins en Traitement Automatique des Langues, est ensuite explicitement interprété comme une méthode d'Apprentissage Automatique. Cette formalisation a permis de mettre en évidence plusieurs de ses éléments caractéristiques. Une particularité notable du modèle réside dans sa capacité à traiter des objets structurés, aussi bien en entrée qu'en sortie, alors que la tâche classique de classification suppose en général un espace de sortie constitué d'un ensemble fini de classes. Nous montrons ensuite comment exprimer le biais d'apprentissage de la méthode à l'aide de l'introduction de la notion d'extension analogique. Enfin, nous concluons par la présentation de résultats expérimentaux issus de l'application de notre modèle à plusieurs tâches de Traitement Automatique des Langues : transcription orthographique/phonétique, analyse flexionnelle et analyse dérivationnelle.

Abstract

The field of Natural Language Processing is mainly covered by two families of approaches. The first one is characterized by linguistic knowledges expressed through rules (production rules for syntax, inference rules for semantics, etc.) operating on symbolic representations. The second one assumes a probabilistic model underlying the data, the parameters of which are induced from corpora of annotated linguistic data. These two families of methods, although efficient for a number of applications, have serious drawbacks. One the one hand, rule-based methods are faced with the difficulty and the cost of constructing high quality knowledge bases: experts are rare and the knowledge of a domain X may not simply adapt to another domain Y. One the other hand, probabilistic methods do not naturally handle strongly structured objects, do not support the inclusion of explicit linguistic knowledge, and, more importantly, heavily rely on an often subjective prior choice of a certain model. Our work focuses on analogy-based methods whose goal is to tackle all or part of these limitations.

In the framework of Natural Language Learning, alternative inferential models in which no abstraction is performed have been proposed: linguistic knowledge is implicitly contained within the data. In Machine Learning, methods with such principles are known as "Lazy Learning". They usually rely on the following learning bias: if an input object Y is "close" to another object X, then its output $f(Y)$ is a good candidate for $f(X)$. Although this hypothesis is relevant for most Machine Learning tasks, the structured nature and the paradigmatic organization of linguistic data suggest a slightly different approach. To take this specificity into account, we study a model relying on the notion of "analogical proportion". Within this model, inferring $f(T)$ is performed by finding an analogical proportion with three known objects X, Y and Z. The "analogical hypothesis" is formalized as: if "X is to Y what Z is to T", then "$f(X)$ is to $f(Y)$ what $f(Z)$ is to $f(T)$". Inferring $f(T)$ from the known $f(X)$, $f(Y)$, $f(Z)$ is achieved by solving the "analogical equation" (with unknown U): "$f(X)$ is to $f(Y)$ what $f(Z)$ is to U".

In the first part of this work, we present a study of this model of analogical proportion within a more general framework termed "analogical learning". This framework is instantiated in several contexts: in the field of cognitive science, it is

related to analogical reasoning, an essential faculty underlying a number of cognitive processes; in traditional linguistics, it gives a support to a number of phenomena such as analogical creation, opposition, commutation; in the context of machine learning, it corresponds to "lazy learning" methods.

The second part of our work proposes a unified algebraic framework, which defines the concept of analogical proportion. Starting from a model of analogical proportion operating on strings (elements of a free monoid), we present an extension to the more general case of semigroups. This generalization directly yields a valid definition for all the sets deriving from the structure of semigroup, which allows us to handle analogical proportions of common representations of linguistic entities such as strings, trees, feature structures and finite sets. We describe algorithms which are adapted to processing analogical proportions of such structured objects. We also propose some directions to enrich the model, thus allowing its use in more complex cases.

The inferential model we studied, firstly designed for Natural Language Processing purposes, can be explicitly interpreted as a Machine Learning method. This formalization makes it possible to highlight several of its noticeable features. One of these characteristics lies in its capacity to handle structured objects, in input as well as in output, whereas traditional classification tasks generally assume an output space made up of a finite set of classes. We then introduce the notion of analogical extension in order to express the learning bias of the model. Lastly, we conclude by presenting experimental results obtained in several Natural Language Processing tasks: pronunciation, flectional analysis and derivational analysis.

Table des matières

CHAPITRE

1 Introduction

1.1 Contexte

La quantité sans cesse grandissante de données linguistiques numérisées disponibles offre de nouvelles perspectives au Traitement Automatique des Langues : l'approche *déductive*, caractérisée par l'application d'une connaissance globale exprimée par exemple sous forme de règles, laisse une place de plus en plus importante à l'approche *inductive*, qui extrait cette connaissance à l'aide de larges ressources linguistiques. Alors que l'approche déductive consiste à inférer *des connaissances particulières à partir de connaissances générales*, l'approche inductive permet d'inférer *des connaissances générales à partir de connaissances particulières*. L'approche inductive en TAL prend principalement la forme de modèles statistiques paramétriques, dont les paramètres sont estimés à l'aide de données ; dans ce contexte, les données servent à induire une abstraction (le modèle paramétré), laquelle peut ensuite s'utiliser de manière déductive sur des nouvelles données à analyser. Le schéma d'inférence alors adopté est le suivant : *particulier* → *général* → *particulier*.

L'approche *analogique* constitue un troisième mode de raisonnement, qui, de même que l'approche inductive, est capable d'exploiter des connaissances particulières contenues dans les corpus de données. Selon ce mode de raisonnement, l'analyse d'une nouvelle entité s'effectue par comparaison avec les données disponibles ; autrement dit, l'inférence s'effectue directement du *particulier au particulier*. Dans cette approche, l'abstraction que constitue la connaissance générale impliquée à la fois dans les approches déductives et inductives n'apparaît plus comme une composante nécessaire du modèle.

Cette approche nous semble intéressante à plusieurs titres : d'un point de vue cognitif, le *raisonnement par analogie* est considéré comme une faculté essentielle, au cœur de nombreux processus cognitifs. Dans un contexte d'Apprentissage Automatique, elle a donné lieu à des méthodes à la fois souples et efficaces ; on parle alors d'*apprentissage paresseux* car l'étape de généralisation (*particulier* → *général*) est évitée. Par ailleurs, cette approche s'accorde bien avec l'*organisation paradigmatique* des données linguistiques, qui permet de mettre aisément une entité linguistique en relation avec d'autres selon des schémas spécifiques ; la connaissance linguistique reste alors implicitement représentée dans le corpus accumulé et les relations

systématiques qu'entretiennent les entités le composant.

En dépit de ces observations, l'utilisation de modèles relevant de cette approche reste marginale en TAL. En outre, les quelques approches que l'on peut qualifier d'analogiques n'exploitent, à notre sens, pas suffisamment (ou pas explicitement) les relations particulières qu'entretiennent les représentations linguistiques. En particulier, l'organisation paradigmatique des données linguistiques invite à considérer des *proportions analogiques*, qui feront l'objet d'une grande partie de ce travail de thèse. Les questions alors soulevées sont : comment exploiter au mieux ces proportions pour effectuer un apprentissage automatique, respectant les principes de l'apprentissage par analogie, et adapté aux tâches de TAL ? comment opérer l'identification des proportions analogiques ?

1.2 Contributions

Les contributions de ce travail sont, à notre sens, multiples. Nous voulons toutefois mettre en exergue trois contributions particulières.

La première contribution est un apport à la méthode d'apprentissage automatique, que nous avons appelée APPA, et qui repose sur l'exploitation de proportions analogiques ; cette méthode est décrite originellement par Pirrelli & Yvon (1999). Notre contribution concerne la caractérisation de cette méthode dans un contexte d'Apprentissage Automatique ; en particulier, nous identifions clairement son biais d'apprentissage à l'aide de l'introduction de la notion d'extension analogique. Nous proposons également des réponses aux problèmes que pose son implantation effective.

La seconde contribution correspond à la proposition d'un cadre général permettant de donner un sens à la notion de proportion analogique. Ce cadre est une généralisation du modèle initialement proposé par Yvon (2003) pour traiter le cas des mots. Cette généralisation permet de caractériser la notion de proportion pour une large gamme de structures algébriques ; en particulier, elle offre un modèle de proportions entre représentations courantes d'entités linguistiques. La cohérence du modèle est appuyée par sa capacité à couvrir correctement des exemples de proportions entre des structures d'apparences différentes.

Enfin, nous avons accordé une place non négligeable à l'implantation d'un système d'apprentissage générique et modulaire. Celui-ci est composé d'un moteur d'apprentissage qui exploite des proportions analogiques ; ce moteur est couplé à des mécanismes indépendants permettant d'identifier des proportions analogiques entre différents types de structures.

1.3 Organisation du document

Ce manuscrit est composé de quatre chapitres principaux, un chapitre introductif, une conclusion et deux annexes.

Le chapitre 2 offre différents points de vue sur l'apprentissage « par analogie ». L'expression par analogie est entre guillemets pour mettre en évidence le fait qu'elle supporte plusieurs acceptions ; seule, elle est ambiguë. Ce chapitre replace cette expression en contexte, et met en relief ce qui lie ou différencie ses acceptions. En particulier, nous étudions le cas du raisonnement par analogie dans le domaine des Sciences Cognitives (section 2.1), dans lequel il est présenté comme une faculté cognitive essentielle. Nous discutons des liens entre ce type de raisonnement et les méthodes d'apprentissage dites paresseuses issues de l'Apprentissage Automatique, et caractérisons spécifiquement ces méthodes (section 2.2). Enfin, nous évoquons la notion de proportion analogique en Linguistique ainsi que l'apprentissage de données à partir d'exemples, dans un contexte de Traitement Automatique des Langues (section 2.3).

Dans le domaine du Traitement Automatique des Langues, de nombreuses tâches s'expriment comme un changement de niveau de représentation. Si l'on dispose de suffisamment de données, il est légitime de vouloir essayer d'apprendre automatiquement ce changement ; le chapitre 3 étudie ce problème d'apprentissage. Après avoir présenté les réponses couramment proposées (section 3.1), nous montrons qu'un apprentissage par analogie, ou plus précisément par exploitation de proportions analogiques, peut offrir des propriétés telles que la gestion naturelle des objets structurés et des ambiguïtés (section 3.2). Par ailleurs, il est connu qu'un apprentissage à partir de données nécessite l'introduction d'un biais d'apprentissage, qui guide l'apprentissage vers des solutions présentant une certaine « forme » ; nous caractérisons le biais de la méthode d'apprentissage décrite à l'aide de la notion d'extension analogique (section 3.3). Enfin, nous montrons l'intérêt d'un tel biais en TAL, en discutant ses liens avec la notion linguistique de création analogique et avec l'organisation paradigmatique des données linguistiques (section 3.4).

La méthode d'apprentissage présentée dans le chapitre 3 a suscité le besoin de savoir caractériser la notion de proportion analogique formelle ; appliquer cette méthode à des représentations linguistiques nécessite de pouvoir identifier des proportions entre ces représentations, et ce, uniquement à partir de leurs formes. Nous présentons dans le chapitre 4 des modèles formels permettant de caractériser cette notion. Nous partons d'un modèle de proportions entre mots, éléments d'un monoïde libre, que nous analysons (section 4.1). Celui-ci est généralisé au cas des semigroupes et des magmas, fournissant un cadre algébrique plus large, qui permet de donner une définition commune à de nombreuses structures (section 4.2). En particulier, cette définition générale s'applique, directement ou après extension, aux représentations habituellement manipulées dans les tâches de TAL, à savoir les structures de traits (section 4.2.3), les langages (section 4.2.4) et les arbres (section 4.2.5), qui recoivent une attention particulière.

Le chapitre 5 regroupe des résultats expérimentaux issus de l'application du modèle d'apprentissage à des tâches que l'on peut exprimer comme des changements de niveau de représentation : prononciation, analyse flexionnelle et analyse dérivationnelle. Les résultats obtenus sont discutés et comparés à un système de

classification de l'état de l'art. Chacune de ces tâches implique des objets de nature différente : chaînes, structures de traits et arbres. Alors que le modèle d'apprentissage que nous défendons peut travailler directement sur les différentes structures, l'approche fondée sur la classification requiert de lourdes reformulations. Nous discutons également de ces problèmes de pré-traitements, qui permettent de mettre en évidence la souplesse de l'approche adoptée.

Nous présentons en conclusion les perspectives que offre ce travail de recherche. Deux annexes accompagnent ce document : une décrivant le logiciel que nous avons développé dans le cadre de cette thèse, une autre présentant des modèles pour lesquels les proportions sont vues comme des similarités de relations.

CHAPITRE

2 Apprendre « par analogie »

There is no word which is used more loosely, or in a greater variety of senses than Analogy.

— John Stuart Mill, *A System of Logic*

And I cherish more than anything else the Analogies, my most trustworthy masters. They know all the secrets of Nature, and they ought to be least neglected in Geometry.

— Johannes Kepler

Mastering the lawless science of our law,
That codeless myriad of precedent,
That wilderness of single instances,
Through which a few, by wit or fortune led,
May beat a pathway out to wealth and fame.

— Lord Alfred Tennyson, *Aylmer's Field*

Sommaire du chapitre

Le terme *apprentissage par analogie* désigne, sous une même appellation, des modèles ou procédés issus de domaines distincts. Le *raisonnement par analogie* correspond à une faculté cognitive essentielle, au cœur de nombreux processus cognitifs : résoudre un problème à l'aide de problèmes déjà résolus, plaider dans un procès à l'aide d'éléments provenant de cas similaires, imiter quelqu'un, reconnaître le lien entre une photo et la situation réelle qu'elle représente, etc. Cette capacité à représenter un objet ou une situation dans un contexte à l'aide d'un objet ou d'une situation rencontrés dans un autre contexte est l'essence même du *raisonnement par analogie*. Ce raisonnement nous permet, en particulier, d'expliquer de nouveaux concepts à l'aide de concepts plus familiers, de décrire des nouveaux phénomènes ou d'adopter une attitude dans une situation inconnue. Il est habituellement modélisé par un appariement entre deux descriptions, correspondant à des situations, la *source* et la *cible* ; une inférence est effectuée par un transfert de connaissance de la situation familière (la source) vers la situation moins familière (la cible), enrichissant de la sorte notre connaissance relative à cette dernière. Un tel type de raisonnement fait l'objet de nombreuses études en Sciences Cognitives (Gentner *et al.*, 2001). Dans ces travaux, l'objectif est de comprendre et de savoir modéliser cette faculté de l'être humain. Étudier le raisonnement par analogie fournit alors un point d'entrée pour la compréhension de mécanismes cognitifs plus généraux.

Le domaine de l'Apprentissage Automatique supervisé regroupe un ensemble de procédés inductifs dont l'objectif est de pouvoir analyser automatiquement un objet à partir d'une base d'objets déjà analysés, comme, par exemple, détecter automatiquement un courriel non sollicité à l'aide d'une base de courriels dont la nature sollicité/non sollicité est déjà connue. Dans ce contexte, l'apprentissage par analogie désigne les méthodes dites *paresseuses*, c'est-à-dire n'effectuant pas de généralisation des données disponibles. Pour analyser un nouvel objet, puisqu'aucune généralisation n'est opérée, le seul recours est de trouver un objet suffisamment « analogue [1] » parmi les objets déjà analysés, et de l'utiliser pour effectuer l'analyse recherchée. L'objet analogue est ici la source et l'objet à analyser la cible. Ces méthodes reposent donc en partie sur les mêmes principes que le raisonnement par analogie tel qu'étudié en sciences cognitives, à savoir l'identification d'un appariement entre deux situations. Toutefois, les objectifs respectifs se distinguent clairement : alors que les uns cherchent à modéliser une faculté cognitive, les autres essaient de résoudre des problèmes d'apprentissage exprimés de manière formelle. Cette formalisation, ainsi que les applications visées, conduisent à considérer des données d'une certaine nature : les données, disponibles et à analyser, sont potentiellement nombreuses, et en général faiblement structurées. À l'inverse, les modèles cognitifs considèrent habituellement un nombre réduit de situations, mais leur modélisation implique des représentations éventuellement complexes. Cette différence de nature des données entraîne également une vision différente de l'appariement caractérisant l'analogie. Dans un cas, il est réduit à une simple mesure de similarité alors qu'il repose sur une conservation de relations et de structures dans l'autre.

1. Cette notion sera formalisée dans la suite.

Dans les domaines de la Linguistique et du Traitement Automatique des Langues, l'approche analogique s'oppose aux modèles opérant une abstraction des données linguistiques, tels que les systèmes à base de règles et les méthodes statistiques paramétriques. Dans les premiers, la connaissance linguistique s'exprime par des représentations symboliques d'entités linguistiques et des règles permettant d'effectuer des traitements sur ces représentations. Les règles peuvent par exemple prendre la forme de règles de production ou de réécritures pour le traitement syntaxique, ou de règles d'inférence pour le traitement sémantique. Ces règles constituent les fondations d'un modèle génératif et abstrait du langage. Les méthodes statistiques paramétriques, quant à elles, font l'hypothèse d'un modèle probabiliste paramétré permettant d'exploiter les données linguistiques. Les paramètres de ce modèle sont induits à partir d'un corpus de données. Une fois ces paramètres estimés, les données ne sont plus manipulées directement : le modèle paramétré représente une abstraction des données, et seule cette abstraction est utilisée. L'approche analogique s'oppose à de telles abstractions. Elle repose à la fois sur des fondements théoriques en linguistique et sur des considérations pratiques en TAL. En linguistique, elle repose sur l'hypothèse suivante : l'analyse des entités linguistiques et de leurs relations peut être apprise et effectuée par analogie avec des entités connues. Selon cette approche, l'abstraction n'apparaît pas comme une composante nécessaire des modèles. En TAL, l'exploitation automatique de données annotées et analysées fournit une alternative intéressante aux approches à base de règles reposant sur des bases des connaissances dont le coût de construction peut se révéler prohibitif.

Notre travail se situe dans ce cadre ; nous étudions comment effectuer un certain nombre d'analyses linguistiques de manière automatique, en exploitant des données représentant des analyses déjà effectuées, et en suivant l'approche analogique. Dans cette optique, il semble légitime de se placer dans un cadre d'apprentissage automatique par analogie, et d'analyser automatiquement des objets par analogie avec d'autres connus, sans construire d'abstraction. Cependant, la nature structurée des objets linguistiques conduit à considérer une vision « pleine » de l'analogie, c'est-à-dire s'écartant de la simple similarité et pouvant modéliser des cas plus complexes.

Ce chapitre a pour objectif de présenter et de mettre en relation les conceptions de l'apprentissage par analogie dans les différents domaines évoqués : les modèles de raisonnement par analogie en sciences cognitives (section 2.1), les apprentis paresseux en apprentissage automatique (section 2.2) et les modèles d'apprentissage du langage par analogie (section 2.3). Nous verrons que, bien que son acception puisse différer selon les domaines, l'apprentissage par analogie repose dans tous les cas sur une caractéristique commune : inférer s'effectue directement à l'aide d'« objets » mémorisés, exemples, cas, situations, entités linguistiques, etc., et n'implique pas d'abstraction de connaissances.

2.1 Les modèles cognitifs de raisonnement par analogie

2.1.1 Introduction

Le *raisonnement par analogie* est une faculté cognitive essentielle. Il est au cœur de nombreux processus cognitifs, qu'il s'agisse par exemple de résoudre un problème à l'aide de problèmes déjà résolus, plaider dans un procès à l'aide d'éléments provenant de cas similaires, imiter quelqu'un, reconnaître le lien entre une photo et la situation réelle qu'elle représente, etc. L'étude de ce type de raisonnement est justifiée et supportée par un large ensemble de résultats empiriques, mettant en évidence les caractéristiques de cette faculté de l'être humain (Gentner *et al.*, 2001).

Analogie et appariements Un tel type de raisonnement fait habituellement appel à un appariement des éléments d'un *domaine source* avec ceux d'un *domaine cible*, domaines a priori différents.

> Analogy will be described as a *mapping* between elements of a *source* domain and a *target* domain.
>
> Hall (1989, p. 40)

En général, les connaissances disponibles au niveau du domaine source sont plus nombreuses que celles relatives au domaine cible. Un des objectifs du raisonnement par analogie est alors d'exploiter un appariement adéquat effectué entre les éléments des domaines source et cible, de manière à transférer un certain nombre de connaissances du premier vers le deuxième.

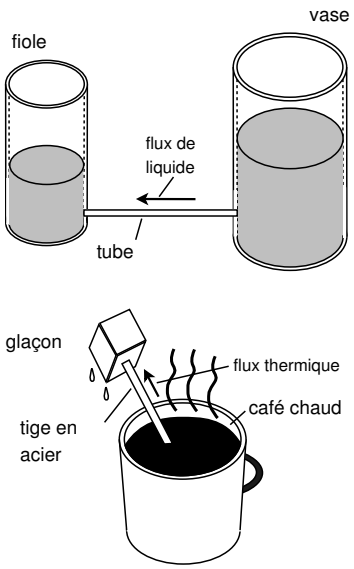


FIG. 2.1 – Analogie entre un flux de liquide et un flux thermique

Par exemple, Falkenhainer *et al.* (1989) examine le cas de l'analogie entre deux configurations, l'une étant le siège d'un flux de liquide (le domaine source) et l'autre d'un flux thermique (le domaine cible). Pour effectuer l'analogie entre les

deux domaines, un certain nombre de connaissances sont supposées accessibles. Le premier domaine est alors représenté par plusieurs objets, tels qu'un vase et une petite fiole remplis d'eau et un tube fin reliant ces deux contenants (cf. figure 2.1). Nous disposons également de certaines propriétés concernant ces objets (la pression exercée sur l'eau dans les deux contenants), et de relations (la pression sur l'eau dans le vase est supérieure à celle contenue dans le fiole, cette différence de pression est la cause d'un flux de liquide du vase vers la fiole). Les informations concernant le second domaine sont du même ordre, mais, par hypothèse, moins complètes. Les éléments le caractérisant sont une tasse contenant du café, un glaçon et une tige d'argent reliant le glaçon au café. Nous savons également que la température du café est supérieure à celle du glaçon. Dans cette situation, raisonner par analogie consiste à apparier la pression, l'eau du vase, l'eau de la fiole avec respectivement la température, le café et le glaçon, conduisant à l'inférence de l'existence d'un flux thermique du café vers le glaçon.

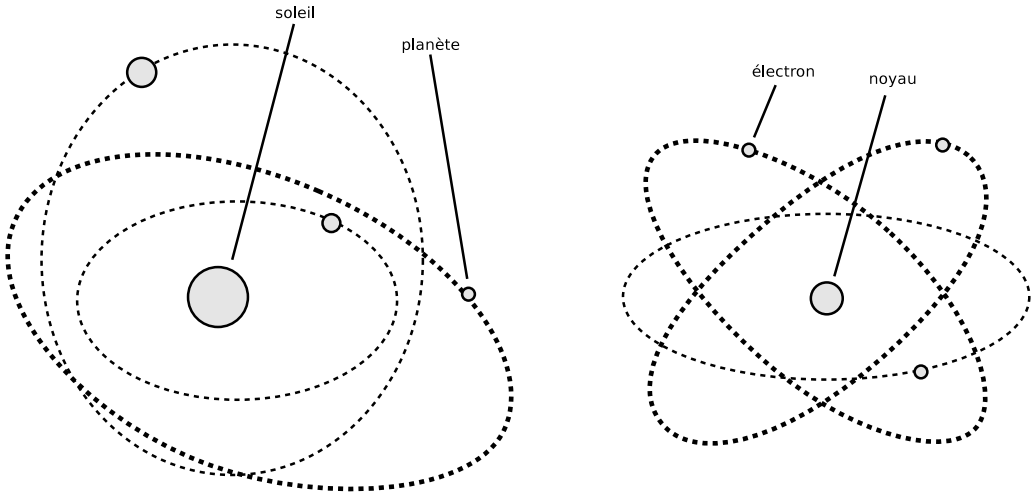


FIG. 2.2 – Analogie de Rutherford

Un tel type de raisonnement est notamment invoqué pour expliquer des cheminements scientifiques pouvant être assez complexes. L'analogie de Rutherford, illustrée par la figure 2.2 est une analogie de la sorte relativement connue. Elle consiste à postuler un « modèle planétaire » pour l'atome, à savoir un noyau très dense de charge positive entouré d'électrons de charge négative se déplaçant dans un volume beaucoup plus important que celui occupé par le noyau. Dans ce modèle, l'atome apparaît alors comme une reproduction miniature du système solaire, qui est composé d'un astre très massif, le soleil, et de planètes moins massives gravitant autour de lui [2]. Cette analogie permet de mettre naturellement en relation les forces d'interaction électromagnétique et gravitationnelle, s'exprimant respectivement par les formules $\vec{F_m} = \mathcal{K}\frac{qq_e}{r^2}\vec{u}$ (loi de Coulomb) et $\vec{F_g} = \mathcal{G}\frac{mm_p}{r^2}\vec{u}$ (loi de Newton). Dans ces formules $\mathcal{G}$ et $\mathcal{K}$ sont des constantes, m, m_p et q, q_e re-

2. Selon ce modèle et d'après la mécanique classique, les électrons gravitant autour du noyau central auraient dû rayonner, donc émettre continuellement des ondes électromagnétiques et, suite à cette perte d'énergie, s'« écraser » sur le noyau. Cette contradiction tombera avec l'adaptation, faite par Niels Bohr, de ce modèle aux travaux de Max Planck et à la théorie quantique de la matière.

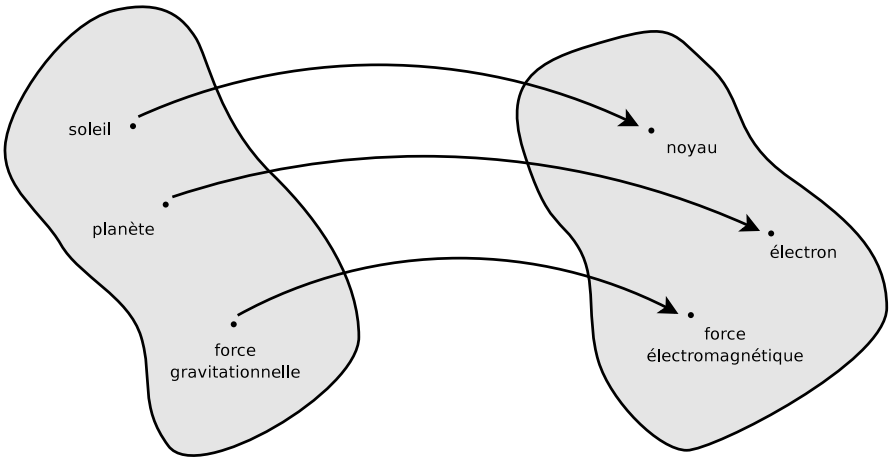


FIG. 2.3 – Analogie de Rutherford : appariement

présentent respectivement les masses du soleil et d'une planète, et les charges du noyau et d'un électron, et r la distance entre les objets considérés [3]. L'appariement effectué dans ce cas est illustré par la figure 2.3.

Le raisonnement par analogie est à la source de nombreuses découvertes scientifiques : Gentner *et al.* (1997) discute certaines des analogies utilisées dans les travaux de Johannes Kepler ; voir aussi Dunbar (2000) et Holyoak & Thagard (1995, chap. 8, the analogical scientist) pour davantage d'exemples.

Analogie et proportions Notons dès à présent que cette conception de l'analogie en tant qu'appariement entre éléments de **deux** domaines différents existe conjointement à une autre de ses acceptions, impliquant quant à elle **quatre** termes formant un *rapport de proportions*, un $\alpha\nu\alpha\lambda o\gamma o\nu$ (*analogon*) en grec. Selon cette acception, l'analogie est une relation entre quatre termes A, B, C, D, exprimée par la proposition « A est à B ce que C est à D » et notée $A : B :: C : D$. Cette conception est celle d'Euclide (Éléments, livre VII, proposition 18 et définition 21), reprise également par Aristote.

> J'entends par « rapport d'analogie » tous les cas où le second terme est au premier comme le quatrième au troisième, car le poète emploiera le quatrième au lieu du second et le second au lieu du quatrième ; et quelquefois aussi on ajoute le terme auquel se rapporte le mot remplacé par la métaphore. Pour m'expliquer par des exemples, il y a le même rapport entre la coupe et Dionysos qu'entre le bouclier et Arès ; le poète dira donc de la coupe qu'elle est « le bouclier de Dionysos » et du bouclier qu'il est « la coupe d'Arès ». De même : il y a le même rapport entre la vieillesse et la vie qu'entre le soir et le jour ; le poète dira donc du soir, avec Empédocle, que c'est « la vieillesse du jour », de la vieillesse que c'est « le soir de la vie » ou « le couchant de la vie ».
>
> Aristote, La Poétique, p 120

3. Ces forces en $\frac{1}{r^2}$ sont quelquefois qualifiées de « newtoniennes ».

Nous parlerons alors de *proportion (analogique)*, et dans le cas évoqué plus haut d'*appariement (analogique)*. Nous dirons également *relation d'analogie* ou simplement *analogie* lorsque le contexte lève toute ambiguïté.

Ces deux conceptions de l'analogie, vue d'une part comme un appariement entre éléments de deux domaines et d'autre part comme un rapport de proportions sont-elles compatibles ? Un rapide examen des exemples précédemment considérés nous permet de répondre par l'affirmative. En effet, dans l'exemple de l'analogie entre les flux de liquide et thermique, l'appariement effectué conduit aux proportions analogiques suivantes : « *la fiole* est au *vase* ce que *le glaçon* est au *café* », « *la pression* est au *flux de liquide* ce que *la température* est au *flux thermique* », etc. Dans le cas de l'analogie de Rutherford, les propositions « *le soleil* est aux *planètes* ce que *le noyau* est aux *électrons* » et « *la masse* est à *la force gravitationnelle* ce que *la charge* est à *la force électromagnétique* » sont tout à fait recevables.

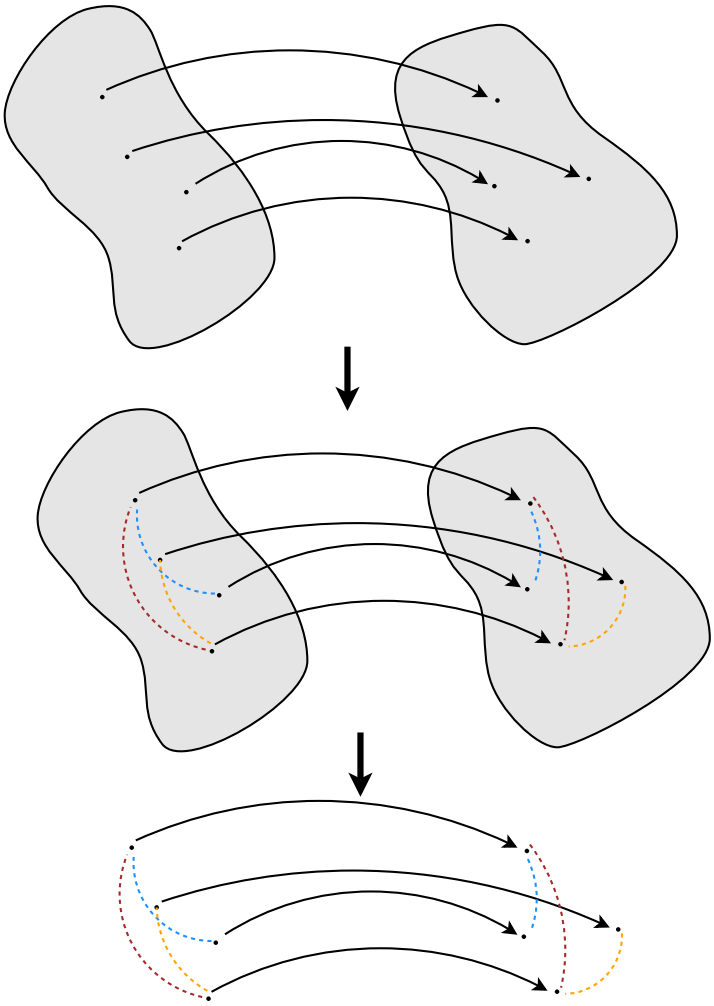


FIG. 2.4 – Proportions analogiques et appariements (a)

Réciproquement, la donnée de telles relations de proportions analogiques permet de « construire » des domaines dont les éléments s'apparient. Ainsi, l'analogie à quatre termes « *la coupe* est à *Dionysos* ce que *le bouclier* est à *Arès* » suggère l'existence de deux « domaines », celui de Dionysos, et celui d'Arès, dont les éléments

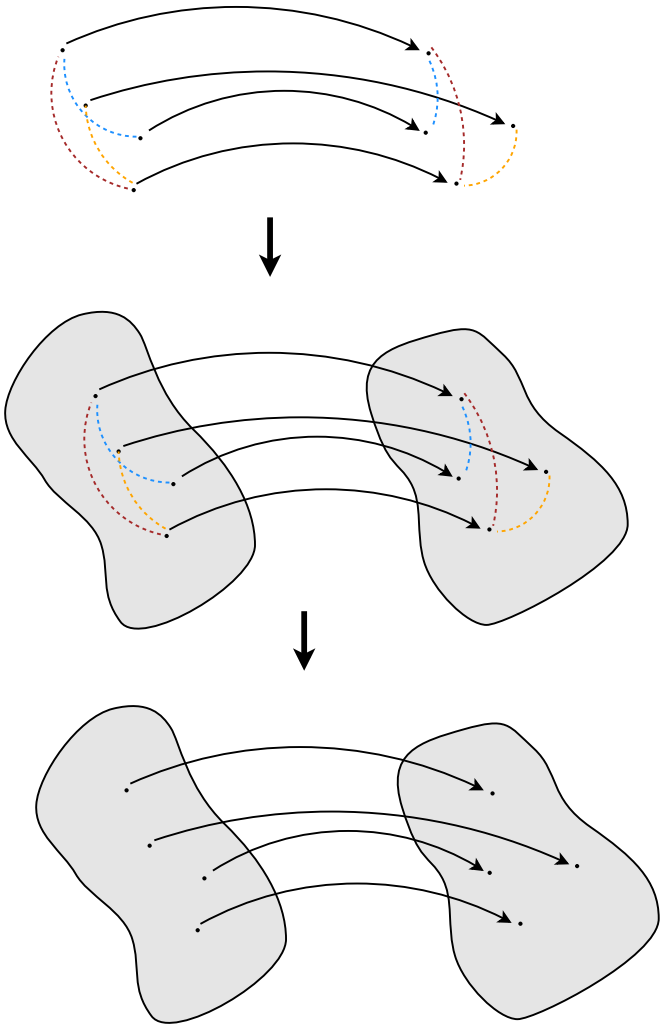


FIG. 2.5 – Proportions analogiques et appariements (b)

peuvent être mis en correspondance. De même, les proportions « *Aphrodite* est à *Arès* ce que *Vénus* est à *Mars* », « *Aphrodite* est à *Athéna* ce que *Vénus* est à *Minerve* », « *Cronos* est à *Athéna* ce que *Saturne* est à *Minerve* », « *Zeus* est à *Éros* ce que *Jupiter* est à *Cupidon* », invitent à considérer l'existence des domaines « divinités de la mythologie grecque » et « divinités de la mythologie romaine ». Ces deux processus sont schématisés sur les figures 2.4 et 2.5. Les deux conceptions évoquées, bien qu'a priori différentes, sont donc en réalité intimement liées. À ce stade, retenons que ces deux acceptions fournissent des modèles d'apparence différente, mais ayant pour base une réalité commune, et qu'il apparaît légitime de les étudier conjointement.

Les étapes principales du raisonnement par analogie

Les modèles de raisonnement par analogie partagent une vision de celui-ci impliquant tout ou partie des étapes représentées sur la figure 2.6 [4].

1. construction d'une *représentation* ;
2. *recherche* (remémoration, accès) d'une situation source analogue ;
3. construction d'un *appariement* entre les éléments des situations source et cible ;
4. *transfert* des éléments inappariés de la source vers la cible (adaptation, apprentissage, inférence).

FIG. 2.6 – Étapes principales du raisonnement par analogie

Durant chacune de ces étapes, des connaissances spécifiques au problème à résoudre peuvent être utilisées pour, par exemple, guider la recherche d'analogues ou améliorer l'appariement entre domaines. Selon les modèles, il est donné plus ou moins d'importance aux différentes étapes, certaines pouvant être complètement occultées. En outre, bien qu'elles soient présentées ici séquentiellement, les modèles peuvent introduire un certain nombre d'interactions entre elles. De manière à identifier plus précisément ces étapes, nous reprenons l'exemple de l'analogie entre flux.

Construction d'une représentation Cette étape consiste à se munir d'un formalisme capable d'exprimer les connaissances disponibles relatives aux deux domaines. Selon les situations, les éléments du domaine peuvent être représentés par des vecteurs d'attributs numériques, symboliques, des éléments d'un formalisme logique, etc. Dans notre exemple, puisque l'on dispose de connaissances sur les relations entre les éléments, un choix possible réside dans l'utilisation d'une logique du second ordre pouvant représenter des termes, des prédicats (d'ordre 1) agissant sur ces termes, et des prédicats (d'ordre 2) s'appliquant à des prédicats d'ordre 1. On a, par exemple, pour le premier domaine :

- *grand(vase), petit(fiole)* ;
- *relié(vase, fiole)* ;
- *supérieur(pression(vase), pression(fiole))* ;
- *cause(supérieur(pression(vase), pression(fiole)), flux)*,

et pour le second :

- *relié(café, glaçon)* ;
- *supérieur(température(café), température(glaçon))*.

Cette phase de construction d'une représentation est fondamentale. Elle pose plus généralement des questions liées à la formalisation de la sémantique des objets considérés. Ce problème de la représentation des connaissances est en réalité

4. Certaines descriptions incluent également les étapes d'évaluation des inférences et d'apprentissage de l'expérience. Nous ne discuterons pas de ces étapes, plus marginales relativement à notre travail.

partagé par toutes les branches de l'Intelligence Artificielle (Kayser, 1997). L'important pour nous ici est de relever que certains modèles prennent cette représentation pour acquise, et que d'autres au contraire cherchent à la faire « émerger » à partir de données plus brutes.

Recherche d'analogues La seconde étape consiste à rechercher dans une mémoire à long terme des situations potentiellement appariables avec la situation à analyser. Cette recherche peut être exhaustive, mais est souvent guidée par des connaissances relatives au contexte. Une position extrême est de ne considérer qu'une seule situation source : c'est ce qui est fait dans l'exemple. Dans un cas plus général, il s'agit d'effectuer cette recherche dans un ensemble potentiellement grand de situations sources. Cela est fréquemment le cas lorsque l'on cherche à résoudre un nouveau problème à l'aide de problèmes déjà résolus. On parle dans ce contexte de *raisonnement à partir de cas* (Kolodner, 1993 ; Aamodt & Plaza, 1994 ; Leake, 1996). Chaque domaine représente alors un couple (problème, solution) ; la recherche d'analogue consiste à repérer les problèmes déjà résolus dont on pense qu'une adaptation au nouveau problème à résoudre serait fructueuse [5]. Par exemple, un garagiste expérimenté pourra identifier une nouvelle situation à des cas déjà résolus, l'amenant à proposer rapidement une solution à un problème mécanique [6].

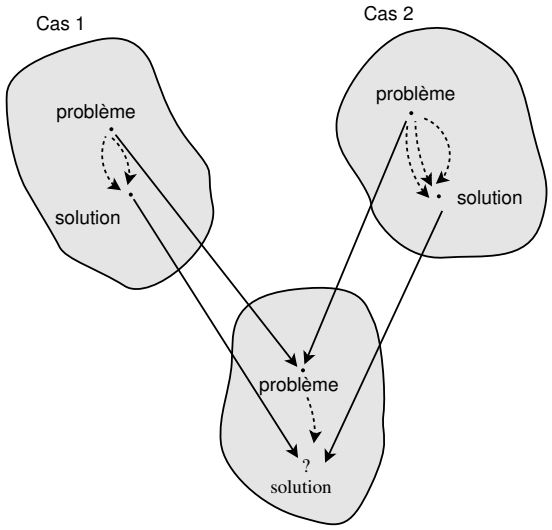


FIG. 2.7 – Raisonnement à partir de cas

Construction d'un appariement La construction d'un appariement a pour objectif de mettre en correspondance les éléments des deux domaines, permettant de

5. « C'est une démarche classique en mathématique : quand on ne parvient pas à démontrer directement un problème posé, on cherche à résoudre un problème analogue. » Gilles Lachaud, Les dossiers de *La Recherche*, n° 20, août-octobre 2005.

6. Le raisonnement à partir de cas repose sur les deux hypothèses suivantes : (i) « des problèmes similaires ont des solutions similaires » ; (ii) « les problèmes futurs risquent d'être similaires aux problèmes courants ».

dire à quel point et en quoi les domaines sont analogues. Cette construction peut se révéler plus ou moins complexe selon la taille des domaines et le mode de représentation adopté. Dans l'exemple de l'analogie entre flux, les termes *café* et *vase* sont appariés, de même que les relations *supérieur(pression(vase), pression(fiole))* et *supérieur(température(café), température(glaçon))*, donnant lieu au schéma représenté sur la partie haute de la figure 2.8. Pour la résolution de problèmes à partir de cas, la construction de l'appariement est directe puisqu'il semble naturel d'apparier les problèmes entre eux et les solutions entre elles.

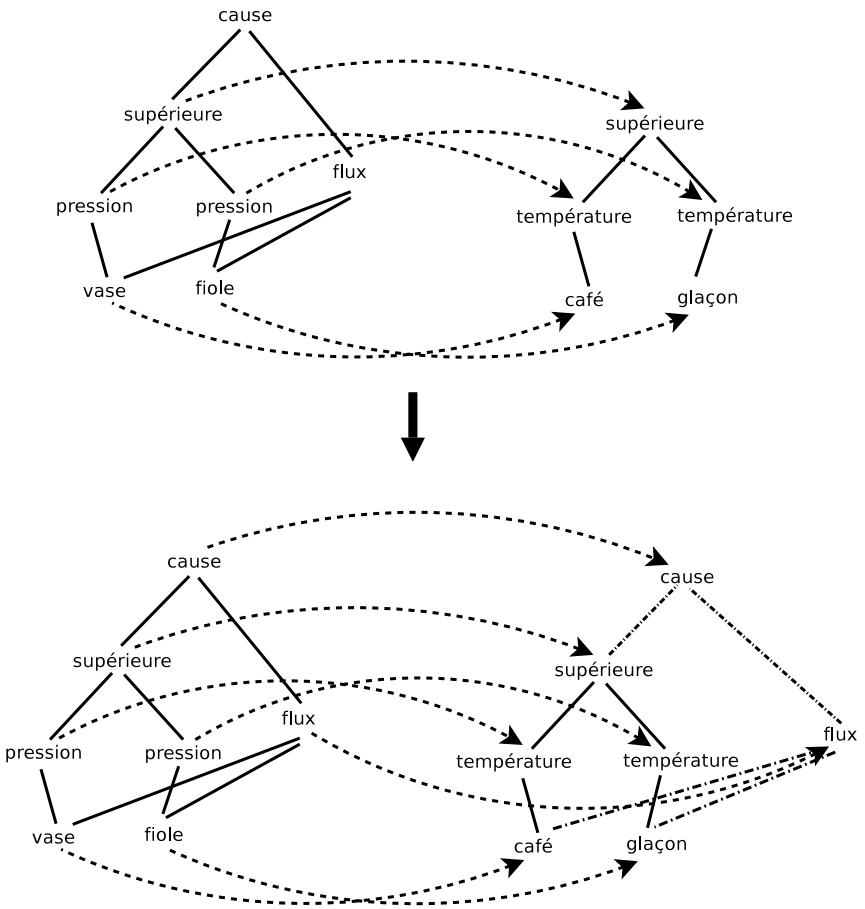


FIG. 2.8 – Appariement et transfert

Transfert L'étape de transfert exploite quant à elle l'appariement effectué de manière à enrichir les connaissances disponibles sur le domaine cible. Elle peut être vue, d'une certaine façon, comme le rétablissement d'un équilibre, consistant à apparier les éléments encore inappariés avec des éléments nouveaux dont l'existence est supposée. Ce processus est illustré sur la partie basse de la figure 2.8. Dans la suite, de manière à différencier les éléments des deux domaines, nous écrirons *élément_1* et *élément_2*. La relation entre les éléments *supérieur_1* et *cause_1* d'une part et l'appariement entre les éléments *supérieur_1* et *supérieur_2* d'autre part, suggèrent : (i) l'existence d'un élément *cause_2*, (ii) une relation entre *supérieur_2* et *cause_2*, (iii) un appariement entre les éléments *cause_1* et *cause_2*. Les

appariements effectués conduisent, en quelque sorte, à une *pression d'existence* sur l'élément *cause_2* et ses liens avec les éléments existants. Cet étape de transfert suppose de savoir compléter un triplet d'éléments (a, a', b) par un quatrième terme b'. Dans notre exemple, le triplet est *(supérieur_1, cause_1, supérieur_2)*, dont la complétion donne *cause_2*. Une telle complétion peut être considérée comme une équation (a, a', b, I), d'inconnue I. Une équation de la sorte sera qualifiée d'*équation analogique*, et sera notée $a : a' :: b :?$ (cf. figures 2.9 et 2.10). Cette notation se comprend aisément lorsque l'on observe que l'équation résolue donne lieu à la proportion analogique $a : a' :: b : b'$, comme nous l'avons fait remarquer précédemment[7]. Cette étape de transfert est, ici aussi, de complexité variable selon les schémas adoptés. Cela explique pourquoi les modèles n'explicitent pas tous le mécanisme de résolution d'équations analogiques. Lorsque ceux-ci impliquent uniquement des équations dont la résolution est triviale, telle que $a : a :: b :?$, ou encore *supérieur_1* : *cause_1* :: *supérieur_2* : *?*, alors l'introduction explicite de la notion n'apparaît pas utile. Cette étape de transfert (ou d'adaptation) est par ailleurs au cœur des systèmes de raisonnement à partir de cas, qui proposent de nombreuses stratégies d'adaptation (Kolodner, 1993), que nous ne détaillerons pas ici.

cf. Sec. 2.1.1, p. 10

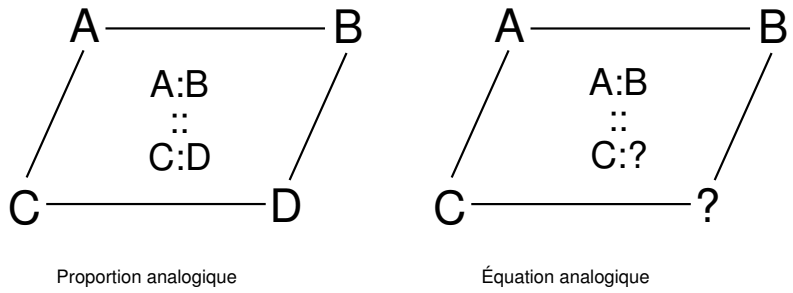


FIG. 2.9 – Proportion et équation analogiques

$$\begin{array}{ccccc} \textit{supérieur\_1} & : & \textit{supérieur\_2} & & \textit{supérieur\_1} & : & \textit{supérieur\_2} \\ & :: & & \rightarrow & & :: & \\ \textit{cause\_1} & : & \textit{?} & & \textit{cause\_1} & : & \textit{cause\_2} \end{array}$$

FIG. 2.10 – Transfert par résolution d'équation analogique

Dans la suite, nous présentons quelques-uns des modèles de raisonnement par analogie tels qu'étudiés en sciences cognitives. Cette présentation n'a aucune prétention d'exhaustivité ; elle est constituée au contraire des quelques modèles les plus pertinents au regard de notre travail. Nous renvoyons à Hall (1989), French (2002), Kokinov & French (2003) et Gentner *et al.* (2001) pour un panorama plus large de l'ensemble de ces modèles. Nous utiliserons les étapes présentées plus haut de manière à faciliter la comparaison des modèles, et à mettre en évidence celles qu'ils privilégient.

Approches symbolique et sub-symbolique L'approche symbolique en Intelligence Artificielle (IA) désigne l'ensemble des modèles dont les mécanismes sous-

7. L'étude de telles proportions et équations analogiques sera l'objet du chapitre 4.

jacents principaux consistent à manipuler des unités et des combinaisons d'unités symboliques et dans lesquels les symboles manipulés sont interprétables (Newell & Simon, 1976 ; Newell, 1980). Les approches sub-symboliques, par exemple connexionnistes, relèvent plutôt d'une démarche ascendante, dans laquelle les représentations émergent à partir de données brutes non interprétées.

L'approche symbolique a dominé pendant une longue période le champ de l'analogie en sciences cognitives. Si l'on accepte de suivre l'idée selon laquelle l'analogie implique un appariement entre relations et une conservation de structures, alors cet état de fait se comprend aisément : l'approche symbolique offre des outils naturels pour manipuler et comparer des objets munis d'une structure relationnelle riche, alors que ces tâches sont restées longtemps problématiques pour les approches sub-symboliques. Cependant, l'importance des représentations structurelles dans les mécanismes cognitifs a conduit à la proposition de modèles sub-symboliques capables de représenter des dépendances relationnelles entre entités, permettant ainsi une modélisation sub-symbolique des processus analogiques.

2.1.2 Les approches symboliques

ANALOGY Le système ANALOGY, proposé par Evans (1968), est la première implantation d'un modèle de résolution d'équation analogique. L'objectif de ce programme est de pouvoir résoudre des équations analogiques géométriques à choix multiples, présentées sous la forme d'une équation analogique « *A* est à *B* ce que *C* est à *?* », où les trois termes *A*, *B* et *C* sont des représentations de figures géométriques. La figure 2.11 donne un exemple de tel problème : à partir des trois figures *A*, *B* et *C*, quelle est la figure *D* choisie parmi les cinq figures du dessous satisfaisant au mieux la relation « *A* est à *B* ce que *C* est à *D* » ? Ces équations proviennent

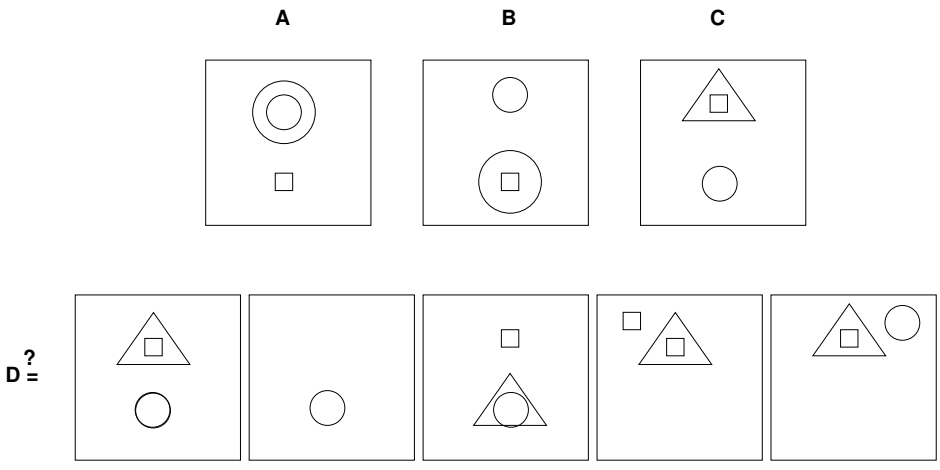


FIG. 2.11 – Exemple d'équation analogique traitée par ANALOGY

en particulier de tests de QI et d'examens tels que les « Scholastic Aptitude Tests » utilisés fréquemment à l'entrée des universités américaines. Une particularité du système ANALOGY réside dans le fait que les termes de l'équation sont présentés sous forme d'une description bas-niveau (points, courbes, etc.).

À partir de ces données, ANALOGY construit ses propres représentations, qualifiées de « haut-niveau » car faisant apparaître des relations entre les éléments des descriptions. Par exemple, la description d'un objet géométrique pourra indiquer qu'un élément p_1 est un carré (noté $carré(p_1)$) et qu'il est « au-dessus » de l'élément p_2 (noté $au\_dessus(p_1, p_2)$), ce second élément pouvant être par exemple un cercle ($cercle(p_2)$). Chaque figure géométrique est alors caractérisée par un ensemble d'objets et de relations entre ces objets. Dans le système ANALOGY, les relations disponibles sont propres au domaine des figures géométriques, à savoir $à\_l'intérieur$, au_dessus, $à\_gauche$, etc. Ce passage bas-niveau → haut-niveau correspond à la première étape de la figure 2.6, à savoir la construction de représentations pour les objets étudiés. Ces représentations ne sont donc pas fournies, mais créées par le système à partir de données plus brutes.

Une fois ces représentations haut-niveau construites, ANALOGY modélise la relation entre les deux figures A et B par l'ensemble des règles permettant de transformer la représentation de A en la représentation de B. Ces règles sont ensuite appliquées à la représentation de C de manière à obtenir des candidats pour D. Une règle est constituée de trois parties : un ensemble d'objets à éliminer, un ensemble d'objets à ajouter, et un ensemble d'objets à apparier entre les deux figures. De plus, un ensemble de transformations (euclidiennes) additionnelles est également disponible. Les candidats proposés pour D sont alors classés selon un certain critère et le meilleur candidat est retenu. Plus précisément, le procédé se décompose en quatre étapes consistant à :

- générer l'ensemble des règles permettant de transformer A en B ;
- générer l'ensemble des règles permettant de transformer C en chacune des solutions proposées ;
- comparer le premier et le deuxième ensembles de règles. Pour chaque paire de règles, il est possible de proposer une généralisation des deux règles, de manière à réconcilier des règles partiellement différentes : plus les règles sont différentes, plus la généralisation à effectuer est importante ;
- choisir la règle transformant C en un choix possible qui généralise le moins la règle transformant A en B, i.e. qui modifie le moins possible la règle originelle permettant de passer de A à B [8].

Ce modèle « simple » possède une caractéristique méritant d'être soulignée dès à présent : la représentation des règles de transformation sous la forme d'ensembles d'opérations consistant à éliminer, ajouter, substituer des éléments offre des similitudes remarquables avec les opérations sous-jacentes aux calculs de distances d'édition entre chaînes de symboles (Wagner & Fischer, 1974). Ce point est discuté un peu plus longuement dans l'annexe A, consacrée en partie à l'étude des systèmes de Lepage et de Miclet, motivés par des objectifs différents de la modélisation de capacités cognitives et exploitant de telles distances d'édition.

8. Nous verrons dans la section 4.2.3 comment formaliser de telles considérations. En particulier, il y sera question de généralisation la plus spécifique.

Théorie de l'appariement structurel La *théorie de l'appariement structurel* (*structure mapping theory*, SMT) de Gentner (1983) constitue une source d'influence incontournable dans l'étude du raisonnement par analogie d'un point de vue cognitif. Le principal objet de cette théorie réside dans l'étude de l'étape 3 de la figure 2.6, c'est-à-dire la construction d'un appariement entre une situation source et une situation cible. Les modèles issus de cette théorie furent les premiers à explicitement souligner l'importance du respect des *structures* dans les domaines source et cible. Plus précisément, un appariement analogique recherche la conservation maximale des *relations* dans les domaines appariés. Ainsi, Gentner (1983) distingue clairement la *similarité*, consistant à apparier des attributs (prédicats d'arité 1), de l'*analogie*, appariant des relations (prédicats d'arité > 1). Dans cette théorie, l'analogie apparaît comme révélatrice d'une ressemblance « de structure », et ne se réduit pas à une similarité « de surface ».

> A theory based on the mere relative numbers of shared and non-shared predicates cannot provide an adequate account of analogy, nor, therefore, a sufficient basis for a general account of relatedness. In the structure mapping theory, a simple but powerful distinction is made among predicate types that allows us to state which ones will be mapped. The central idea is that an analogy is an assertion that a relational structure that normally applies in one domain can be applied in another domain. [...] Overlap in both object-attributes and inter-object relationships is seen as literal similarity, and overlap in relationships but not objects is seen as analogical relatedness.
>
> Gentner (1983, p. 156,161)

> The central idea is that an analogy is a mapping of knowledge from one domain (the base) into another (the target) which conveys that a system of relations known to hold in the base also holds in the target. The target objects do not have to resemble their corresponding base objects. Objects are placed in correspondence by virtue of corresponding roles in the common relational structure. [...] This structural view of analogy is based on the intuition that analogies are about relations, rather than simple features.
>
> Falkenhainer *et al.* (1989)

Reprenons l'exemple évoqué plus haut de l'analogie entre un flux de liquide et un flux thermique. *cf. Sec. 2.1, p. 8*

À l'inverse du système ANALOGY créant ses propres représentations haut-niveau, la théorie de l'appariement structurel suppose la connaissance relative aux domaines déjà fournie sous forme relationnelle. Les assertions suivantes sont donc directement exploitables :

- *grand(vase)*, *petit(fiole)*, *noir(fiole)* ;
- *relié(vase, fiole)* ;
- *supérieur(pression(vase), pression(fiole))* ;
- *cause(supérieur(pression(vase), pression(fiole)), flux)* ;
- *relié(café, glaçon)* ;
- *noir(café)* ;
- *petit(glaçon)* ;
- *supérieur(température(café), température(glaçon))*.

Formellement, la théorie de l'appariement structurel représente un domaine par un ensemble de termes et de prédicats dans une logique de second ordre. Dans les assertions posées, *vase*, *fiole*, *café* et *glaçon* sont des termes (constants). Les expressions *pression(vase)* et *température(glaçon)* sont également des termes, créés à partir d'un foncteur (par ex. *pression*) et d'un autre terme (par ex. *vase*). Les prédicats se distinguent par leur *arité* et leur *ordre*. L'arité est le nombre d'arguments attendu par le prédicat, par exemple 2 pour *supérieur(pression(vase), pression(fiole))* et 1 pour *grand(vase)*. Un prédicat s'appliquant à un terme, comme *noir(café)*, est d'ordre 1. Un prédicat s'appliquant à un prédicat d'ordre 1 est d'ordre 2 ; c'est le cas de *cause(supérieur(pression(vase), pression(fiole)), flux)*. Nous appellerons *attributs* les prédicats d'arité 1 et d'ordre 1, et *relations* les autres prédicats.

L'appariement consiste à mettre en correspondance des termes (par exemple *vase* → *café*) et des prédicats (par exemple *supérieur(pression(vase), pression(fiole))* → *supérieur(température(café), température(glaçon))*).

Pour effectuer l'appariement, la théorie de l'appariement structurel expose un certain nombre de critères à considérer : (i) les appariements entre relations doivent être préférés aux appariements entre attributs ; (ii) seules des relations identiques peuvent être appariées ; (iii) les appariements impliquant des relations d'ordre supérieur sont prioritaires. Ce dernier critère correspond au principe de *systématicité* : on recherche la mise en correspondance de systèmes de relations et non de relations isolées. Cette théorie est implanté dans le système SME (the Structure-Mapping Engine) (Falkenhainer *et al.*, 1989 ; Forbus *et al.*, 1994), pouvant être couplé au système de recherche d'analogues MAC/FAC (Gentner & Forbus, 1991 ; Forbus *et al.*, 1995).

cf. Sec. 2, p. 6

Nous avions suggéré, dans l'introduction, qu'un appariement analogique impliquait la présence de proportions analogiques. Nous montrons ici que les critères envisagés par la théorie de l'appariement structurel sont cohérents avec cette suggestion. Tout d'abord, mettre en correspondance deux prédicats d'arité supérieure à 1 revient à créer des proportions entre leurs arguments. Ainsi, si $p(x_1, \ldots, x_n)$ est apparié avec $q(y_1, \ldots y_n)$, on a $x_1 : x_2 :: y_1 : y_2$, et plus généralement, $x_i : x_j :: y_i : y_j$ pour tout couple (i, j) tel que $i \neq j$. Ensuite, le principe de systématicité, tendant à conserver la structure globale des systèmes, appuie la validité de telles proportions.

Schmid Le modèle proposé par Schmid *et al.* (2003) retiendra également notre attention. Nous en discutons dans la section 4.2.4.

2.1.3 Les approches sub-symboliques

Plate Plate (2000) propose un modèle sub-symbolique d'analogie fondé sur les *Holographic Reduced Representation* (Plate, 1995), capables de représenter les relations, et donc les structures, dans un vecteur « plat » de taille fixe. Le modèle repose

sur l'exploitation de l'opération de convolution, $\otimes$, définie par :

$$\forall x, y, z \in \mathbb{R}^n, z = x \otimes y \Leftrightarrow \forall i \in [\![1, n]\!], z_i = \sum_{k=1}^{n} x_k y_{i-k \mod n}.$$

Cette opération est associative, commutative, possède un élément neutre, et un inverse existe pour la plupart des vecteurs. Le vecteur $\bar{x}$ défini par $\bar{x}_i = x_{-i \mod n}$ est une approximation de cet inverse s'il existe. Cette opération de convolution permet de modéliser le résultat de l'« association » entre deux éléments x et y par $x \otimes y$. Il est ensuite possible de considérer plusieurs associations, par la simple addition : $z = x \otimes y + v \otimes w$. Pour retrouver le terme associé à x dans z, z est convolé avec $\bar{x}$, ce qui donne approximativement y, si $\bar{x} \otimes v$ est petit devant $\bar{x} \otimes x$.

Pour l'analogie flux de liquide/flux thermique, tous les termes et les prédicats (*vase*, *pression*, etc.) sont représentés par des vecteurs de $\mathbb{R}^n$ tirés aléatoirement selon une certaine loi. Pour modéliser les relations entre ces éléments, on pose ensuite :

- *PV* = *vase* + *pression* $\otimes$ *vase* ;
- *PF* = *fiole* + *pression* $\otimes$ *fiole* ;
- *PVPF* = *supérieur* + *supérieur_1* $\otimes$ *PV* + *supérieur_2* $\otimes$ *PF*,

Dans cette représentation, *supérieur_1* (resp. *supérieur_2*) correspond au premier (resp. second) argument du prédicat *supérieur* ; *PV* est un nouvel élément, dont le rôle est de rendre compte de la relation entre la pression (*P*) et le vase (*V*). La propriété principale du modèle est que tous les vecteurs impliqués ont la même taille n : la complexité des objets est conservée à l'ajout de relations.

Cette représentation étant donnée, l'analogie entre deux situations est simplement modélisée pour Plate par un produit scalaire entre vecteurs, c'est-à-dire une mesure de similarité. Puisque les vecteurs tiennent compte de la structure des situations, cette similarité peut être vue comme une similarité structurelle. Ceci nous permet de dire à quel point deux situations sont analogues, mais n'exhibe pas d'appariement. Eliasmith & Thagard (2001) présentent un modèle (DRAMA), fondé sur les HRR de Plate, et traitant spécifiquement du problème de l'appariement pour de telles représentations. Notons que la taille des vecteurs requise pour que le système soit en mesure de fonctionner dépend clairement du nombre de relations que l'on superpose. Le comportement d'un tel modèle face à un nombre élevé d'éléments fortement structurés ne nous apparaît pas clair [9].

2.1.4 Les modèles hybrides

L'équipe d'Hofstadter Hofstadter et les membres de son équipe critiquent, quant à eux, l'approche consistant à étudier indépendamment les processus perceptuels (dits bas-niveau) et les processus conceptuels (dits haut-niveau). Ils s'opposent ainsi à une attitude largement partagée dans le domaine de l'Intelligence Artificielle symbolique traditionnelle (Newell & Simon, 1976 ; Newell, 1980), qui justifie l'étude exclusive des processus conceptuels par cette propriété d'indépendance.

9. Plus précisément, il s'agirait de mettre clairement en relation le nombre d'éléments et la profondeur des relations à la taille des vecteurs.

> Much work in artificial intelligence has attempted to model conceptual processes independently of perceptual processes, but we will argue that this approach cannot lead to a satisfactory understanding of the human mind. [...] While this objectivist position has been unfashionable for decades in philosophical circles (especially after Wittgenstein's work demonstrating the inappropriateness of a rigid correspondence between language and reality), most early work in AI implicitly accepted this set of assumptions.
>
> Chalmers *et al.* (1995, p. 170)

Leur modèle est à l'inverse fondé sur la notion de perception de haut-niveau (*high-level perception* en anglais). Cette perception est une fonction cognitive au centre du modèle, car elle permet de modéliser les interactions entre les processus perceptuels et conceptuels. Cette vision est, en quelque sorte, à rapprocher de l'attitude kantienne.

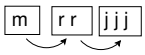
> Sans la sensibilité, nul objet ne nous serait donné et sans l'entendement nul ne serait pensé. Des pensées sans contenu (Inhalt) sont vides, des intuitions sans concept, aveugles.
>
> Kant (Critique de la Raison Pure, p. 76-77)

Pour Hofstadter et ses co-auteurs, il est impossible de considérer séparément les « modules » perceptuels et conceptuels ; ils se nourrissent l'un l'autre. Le raisonnement par analogie, auquel Hofstadter accorde une large place dans l'étude des processus cognitifs, doit rendre compte de cette interaction. Ainsi, les étapes impliquées dans le raisonnement par analogie, comme celles présentées sur la figure 2.6, ne peuvent correspondre à une succession de processus invoqués séquentiellement ; les processus interagissent et opèrent en parallèle. Par exemple, la première étape, consistant à construire une représentation, n'est pas dissociable du reste.

> [S]eparating perception from the "higher" tasks for which it is to be used is almost certainly a misguided approach. The fact that representations have to be adapted to particular contexts and particular tasks means that an interplay between the task and the perceptual process is unavoidable, and therefore that any "modular" approach to analogy-making will ultimately fail. It is therefore essential to investigate how the perceptual and mapping processes can be integrated. [...] The hand-coding of representations is endemic in traditional AI. Any program that uses pre-built representations for a particular task could be subject to such a "representation module" argument similar to that given above. For most purposes in cognitive science, an integration of task-oriented processes with those of perception and representation will be necessary.
>
> Chalmers *et al.* (1995, p. 189)

Ces considérations sont implantées dans le programme COPYCAT (Hofstadter & Mitchell, 1995 ; Mitchell, 2001). Celui-ci traite principalement de la résolution d'équations analogiques complexes sur des séquences de symboles, telles que « *abc* est à *abd* ce que *ijk* est à *quoi ?* », « *aabc* est à *aabd* ce que *ijkk* est à *quoi ?* » et « *abc* est à *abd* ce que *mrrjjj* est à *quoi ?* ». Les auteurs de ce système ont délibérément décidé de ne travailler que sur le « micro-domaine » des séquences de lettres. La raison en est que de tels objets sont à la fois simples mais potentiellement porteurs de structure. En outre, les auteurs accordent une certaine « universalité » à de tels objets (Hofstadter & Mitchell, 1995, p. 208).

Pour résoudre de tels problèmes, COPYCAT ne fait pas d'hypothèse a priori sur la nature des structures ou sur les opérateurs à utiliser. Il repose au contraire sur l'émergence dynamique de structures perceptuelles haut-niveau. Une telle émergence repose sur un mécanisme stochastique de recherche contrôlé par un facteur de température. La séquence *mrrjjj* peut ainsi donner lieu à la représentation structurée suivante :



permettant de proposer *mrrkkk* comme solution à l'équation *abc* : *abd* :: *mrrjjj* :?. Cette représentation n'est pas fixée a priori, mais fait partie intégrante du processus global de résolution d'équation analogique. Nous reviendrons succinctement sur ce modèle dans la section 4.2.4.

Citons également le système de recherche d'analogues ARCS (Thagard *et al.*, 1990), qui peut être couplé avec le système ACME (Holyoak & Thagard, 1989), dans lequel l'analogie correspond à un processus émergeant de l'activation d'états d'une architecture connectionniste. Voir également Hummel & Holyoak (1997) et leur système LISA.

Remarque Les modèles étudiés permettent de mettre en évidence plusieurs caractéristiques de l'apprentissage par analogie. Nous retiendrons en particulier qu'il implique une **inférence effectuée directement à partir d'exemples** (sans généralisation), et **reposant sur un appariement structurel donnant naturellement lieu à des proportions analogiques**.

2.2 Les apprentis paresseux en apprentissage automatique

2.2.1 Introduction

L'Apprentissage Automatique (*Machine Learning* en anglais) désigne un ensemble de procédés inductifs dont l'objectif est d'« apprendre » à partir de données. Un scénario d'apprentissage courant consiste à chercher à prédire une information non connue sur un objet représenté par un certain nombre de caractéristiques appelées *attributs*. Il peut s'agir, par exemple, d'évaluer le risque d'attaque cardiaque d'un patient dont on connaît quelques données cliniques (taille, poids, antécédents médicaux, etc.). Dans ce cas, les données cliniques représentent les attributs du patient, et le risque d'attaque correspond à l'information à prédire. De manière à effectuer cette prédiction, les méthodes d'apprentissage automatique exploitent les données de patients dont le risque d'attaque cardiaque est déjà connu. Citons également les tâches de détection automatique du caractère non-sollicité d'un courrier électronique (i.e. le classer en pourriel ou non-pourriel) et de reconnaissance automatique de chiffres manuscrits (i.e. associer à une image le chiffre qu'elle représente).

La tâche d'apprentissage automatique (supervisé) se formalise de la façon suivante. Étant donnée une base d'apprentissage, constituée d'objets représentés par un ensemble d'attributs et dont la valeur de l'information à prédire est connue, il s'agit de construire un apprenti en mesure de prédire l'information manquante d'un objet non encore rencontré (i.e. absent de la base d'apprentissage). En notant x_i un objet et y_i la valeur de l'information cible, la *base d'apprentissage* BA est constituée de couples $(x_i, y_i) \in X \times Y$:

$$BA = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subset X \times Y.$$

L'espace X (resp. Y) est l'*espace d'entrée* (resp. *de sortie*). Un *apprenti* est une fonction $f : X \to Y$, qui à tout $x \in X$ associe une valeur de prédiction $f(x) \in Y$. Une méthode d'apprentissage automatique est un algorithme qui construit un apprenti $f \in Y^X$ à partir d'une base d'apprentissage BA. La *phase d'apprentissage* désignera la période de construction de cet apprenti. Nous parlerons également de *phase de test* pour désigner la période consistant à tester et à appliquer l'apprenti sur des données nouvelles [10]. Mitchell (1997), Hastie *et al.* (2001) et Duda *et al.* (2000) offrent un panorama de la variété des réponses à ce problème, de même que Cornuéjols & Miclet (2002) en français.

Quand les sorties prennent leurs valeurs dans un ensemble fini, on parle généralement de *classification* ; les sorties sont alors appelées classes. Dans la suite, pour illustrer les méthodes d'apprentissage, nous nous placerons dans le cadre de la classification binaire [11] (deux classes), et de manière à simplifier les calculs, nous poserons $Y = \{-1, +1\}$. De plus, remarquons que, dans ce cas, à partir d'une fonction $g : X \to \mathbb{R}$, il est toujours possible de construire un apprenti f en posant $\forall x, f(x) = sign(g(x))$. La construction d'une fonction $g \in \mathbb{R}^X$ est donc une procédure plus générale [12]. Dans la suite, nous pourrons donc considérer des fonctions de $\mathbb{R}^X$ pour répondre à des problèmes de classification binaire. Nous noterons g de telles fonctions et f l'apprenti associé.

Les k plus proches voisins Illustrons le schéma général d'apprentissage automatique présenté ci-dessus par la présentation d'un algorithme, l'algorithme des k plus proches voisins (k-ppv) (Fix & Hodges, 1951 ; Cover & Hart, 1967 ; Dasarathy, 1991). L'algorithme des k-ppv est simple, ce qui ne l'empêche pas de se révéler redoutablement efficace face à de nombreux problèmes. Son principe est le suivant : pour classer un objet inconnu x, on recherche les k plus proches voisins de x contenus dans la base d'apprentissage, et on affecte à x la classe la plus représentée chez

10. Nous avons décrit ici le problème de l'apprentissage automatique *supervisé* : celui-ci est dit supervisé car, sur les exemples de la base d'apprentissage, l'information à prédire est connue. Nous n'aborderons pas d'autres cadres tels que l'apprentissage automatique non-supervisé ou l'apprentissage par renforcement. Dans la suite, et sauf mention contraire, apprentissage automatique signifiera (abusivement) apprentissage automatique supervisé.

11. Un certain nombre de méthodes permettent en outre de reformuler un problème d'apprentissage multi-classes en un problème de classification binaire.

12. Elle permet généralement d'associer une certaine mesure de confiance à la prise de décision d'un apprenti ; elle exprime alors que la décision $f(x) = 1$ est effectuée de manière plus sûre avec $g(x) = 0,9$ qu'avec $g(x) = 0,2$.

ses voisins. Dans un problème de classification binaire, avec $y_i \in \{-1, +1\}$, cela s'exprime simplement par :

$$g(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i, \tag{2.1}$$

où $N_k(x)$ représente l'ensemble des k plus proches voisins de x dans BA. Une illustration de cet algorithme est donnée sur la figure 2.12, dans laquelle il s'agit de classer un point selon les classes *rond* et *croix*. Le point inconnu sur la gauche se voit attribuer la classe *croix* car deux parmi ses trois plus proches voisins sont des croix. Le même raisonnement permet d'affecter la classe *rond* au point inconnu sur la droite. Est représentée également sur les figures 2.13 et 2.14 la *frontière de séparation* pour deux valeurs de k : 1 et 9. Cette frontière sépare les surfaces correspondant à la décision de l'algorithme en faveur d'une certaine classe.

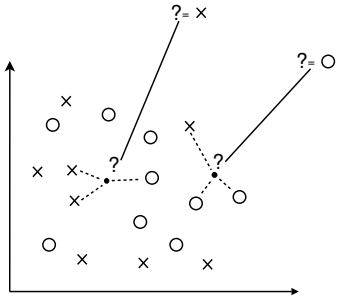


FIG. 2.12 – Principe de l'algorithme des k-plus proches voisins, $k = 3$

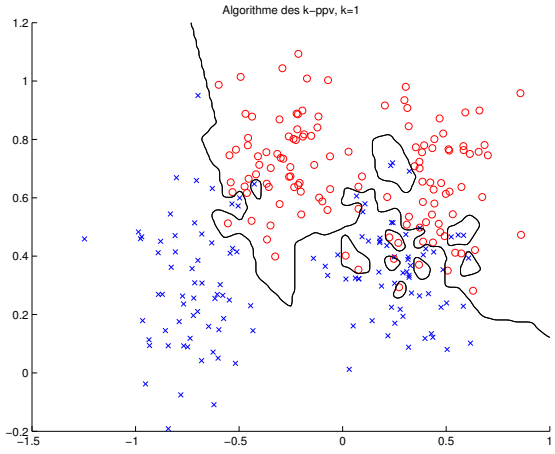


FIG. 2.13 – Algorithme des k-plus proches voisins, $k = 1$

Hypothèse de modèle, recherche et minimisation L'algorithme des k-ppv fournit une réponse pratique au problème de la classification automatique. Formellement, ce problème peut être modélisé par une recherche dans l'espace des fonctions G de X dans Y ($G = Y^X$). Cette recherche est guidée par la base d'apprentissage $BA \subseteq X \times Y$, et aboutit à la proposition d'un apprenti $g \in G$. Cette recherche correspond à ce que nous avons désigné plus haut par phase d'apprentissage.

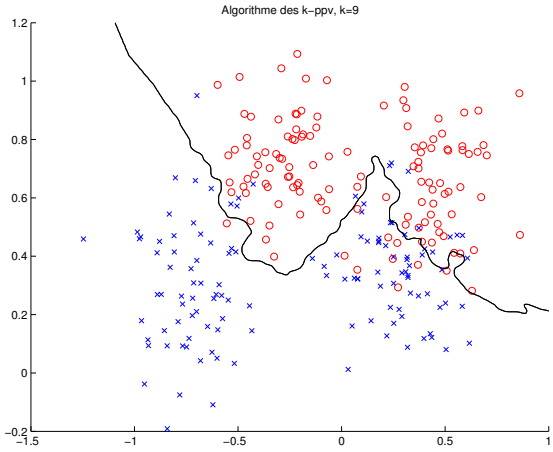


FIG. 2.14 – Algorithme des k-plus proches voisins, $k = 9$

On considère généralement que les exemples de la base d'apprentissage sont distribués indépendamment et identiquement selon une certaine distribution inconnue $\mathbb{P}(X, Y)$. L'objectif de la recherche est alors de trouver un apprenti $g \in Y^X$ minimisant l'*erreur de classification*, à savoir la quantité

$$L(g) = \mathbb{P}\{g(X) \neq Y\} = \mathbb{E}\{\mathbb{1}_{[g(X)\neq Y]}\},$$

où $\mathbb{1}$ représente la fonction indicatrice.

Puisque la distribution $\mathbb{P}$ est inconnue, plusieurs critères peuvent être envisagés de manière à minimiser cette quantité. Une approche consiste à :

- réduire l'espace de recherche en se limitant à un ensemble d'apprentis $H \subset G$. Cet ensemble correspond à une *hypothèse de modèle* ;
- spécifier une fonction de coût dépendant des données permettant de discerner les « bons » apprentis et ainsi guider la recherche.

Les séparateurs linéaires Par exemple, si $X = \mathbb{R}^d$ et $Y = \{-1, 1\}$, l'espace H des apprentis acceptables peut regrouper l'ensemble des séparateurs linéaires, i.e.

$$H = \{g_w, w \in \mathbb{R}^d | \forall x \in X, g_w(x) = x^t.w\}.$$

Le risque quadratique empirique (principe des moindres carrés) est une fonction de coût standard [13] :

$$L_{emp}(g_w) = \frac{1}{n}\sum_{i=1}^{n}(g_w(x_i) - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i^t.w - y_i)^2.$$

L'objectif de la recherche est alors de trouver l'apprenti minimisant $L_{emp}(g_w)$,

$$g^*_{emp} = argmin_{g_w \in H} L_{emp}(g_w).$$

13. Ce risque n'est pas égal à l'erreur empirique ($\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{[sign(g_w(x_i))\neq y_i]}$), mais il apparaît plus naturel dans le cas des séparateurs linéaires.

Dans cette situation, la solution du problème de minimisation est directement explicitable. En notant $X = (x_1, \ldots, x_n)$ (matrice $n \times d$) et $y = (y_1, \ldots, y_n)$, on a, sous la condition que $(X^t X)^{-1}$ existe :

$$argmin_{g_w \in H} L_{emp}(g_w) = (X^t X)^{-1} X^t y, \text{ soit}$$

$$\forall x, g^*_{emp}(x) = x^t (X^t X)^{-1} X^t y.$$

Pour les mêmes données que celles utilisées sur les figures 2.13 et 2.14, le meilleur séparateur linéaire est illustré sur la figure 2.15.

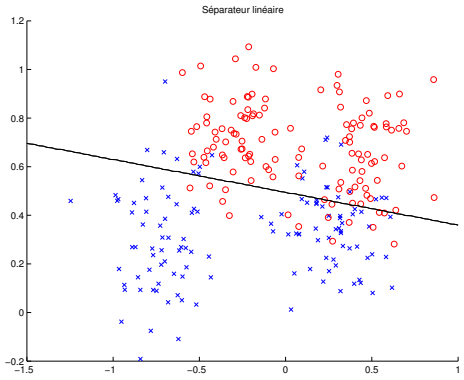


FIG. 2.15 – Séparateur linéaire

2.2.2 Généralisation vs. paresse

Cet exemple simple nous permet de différencier les deux réponses fournies au problème de la classification binaire, à savoir l'algorithme des k-ppv et la séparation linéaire. Cette dernière fait une hypothèse de modèle forte : elle considère que les données présentent une propriété de séparabilité linéaire globale. Cette hypothèse se caractérise par une réduction explicite de l'espace des fonctions recherchées. Une telle hypothèse de modèle prend généralement la forme d'une famille paramétrée de fonctions. Nous parlerons alors d'*apprentissage paramétrique*. Dans le cas de la séparation linéaire, l'ensemble des fonctions de H est paramétré par le vecteur w. La tâche d'apprentissage consiste alors à trouver le paramètre minimisant une certaine fonction de coût. Se doter de ces deux ingrédients, une hypothèse de modèle et un coût à minimiser, correspond à l'approche évoquée au paragraphe 2.2.1. Un apprenti construit de la sorte est complètement formulable : il est entièrement déterminé par la donnée du paramètre w, comme ce fut le cas dans l'exemple des séparateurs linéaires : $\forall x, g^*_{emp}(x) = x^t w$ avec $w = (X^t X)^{-1} X^t y$. Par ailleurs, une fois effectuée la phase d'apprentissage consistant à inférer ce paramètre sur les données, celles-ci peuvent être complètement oubliées. Elles sont inutiles puisque seul le paramètre w est nécessaire. Nous dirons alors qu'une *généralisation* a été effectuée : le paramètre w généralise les données de la base d'apprentissage. Le mécanisme d'inférence se décompose ainsi en deux temps : (i) construction d'une généralisation, (ii) application de la généralisation à des données nouvelles.

L'algorithme des k-ppv présente des caractéristiques tout à fait différentes ; l'implicite prend le pas sur l'explicite. Dans cet algorithme, même si celle-ci n'a pas été formulée par la donnée d'un modèle, il existe bien une hypothèse sous-jacente au mécanisme d'apprentissage. Elle correspond au principe suivant : « si a est proche de b, alors $f(a)$ est un bon candidat pour $f(b)$ », ce qui revient à supposer une constance locale de la fonction visée. Une hypothèse est donc effectuée, mais contrairement au cas de la séparation linéaire, elle ne prend pas une forme explicite. Cet algorithme ne fait pas d'hypothèse de modèle, et ne recherche pas explicitement à minimiser un coût, contrairement au cas paramétrique [14]. Dans l'algorithme des k-ppv, l'apprenti n'est pas réellement « construit », et la phase d'apprentissage se résume au simple stockage des exemples de la base d'apprentissage. Cette phase est donc très peu coûteuse (en temps). En contrepartie, telle la cigale dans la fable, l'algorithme sera soumis à davantage d'efforts lors de la phase de classification, puisque, pour chaque nouvel exemple à analyser, une recherche de voisins dans la base d'apprentissage doit être effectuée [15]. C'est la raison pour laquelle nous parlerons d'*apprentissage paresseux* (*lazy learning* en anglais) (Aha *et al.*, 1991 ; Aha, 1997 ; Mitchell, 1997). D'autres termes sont fréquemment utilisés pour désigner des méthodes reposant sur des principes similaires : apprentissage à partir de cas, à partir d'exemples, à partir d'instances, à partir d'exemplaires, fondé sur la mémoire, etc ; dans la suite, nous parlerons d'apprentissage paresseux et n'opérerons aucune distinction supplémentaire. Ces méthodes reposent sur une connaissance restant implicitement représentée dans les exemples accumulés de la base d'apprentissage, et s'opposent par conséquent à la construction d'une généralisation explicite des données. L'inférence s'effectue directement par analogie avec les cas connus. L'ensemble de ces méthodes constituera donc pour nous la référence de l'apprentissage par analogie dans le contexte de l'Apprentissage Automatique.

2.2.3 Apprentissage paresseux et raisonnement par analogie

Les méthodes d'apprentissage paresseux n'effectuent pas de généralisation des données disponibles. Pour analyser un nouvel objet, aucune généralisation n'est opérée, et le seul recours consiste à trouver un objet suffisamment « analogue » parmi les objets déjà analysés, et de l'utiliser pour effectuer l'analyse. En utilisant le vocabulaire introduit dans la section 2.1, l'objet analogue trouvé tient lieu de source et l'objet à analyser de cible. Ces méthodes reposent donc en partie sur les mêmes principes que le raisonnement par analogie étudié dans le contexte des sciences cognitives, à savoir la recherche et l'identification d'un appariement entre deux situations. Ces observations justifient la considération selon laquelle un apprentissage paresseux est un apprentissage par analogie. L'algorithme des

14. Cette remarque n'est pas en contradiction avec le fait qu'on dispose d'un certain nombre de résultats énonçant les capacités de généralisation des k-ppv. Construire un algorithme ayant pour objectif explicite la minimisation d'une fonction, et étudier a posteriori les propriétés d'un algorithme relèvent bien de deux démarches distinctes.

15. En revanche, contrairement à la cigale, il n'ira crier famine nulle part…

k-ppv peut, par exemple, être réexprimé selon les étapes décrites sur la figure 2.6, comme l'illustre le tableau 2.1.

Raisonnement par analogie	***k*-ppv**
Construction d'une représentation	Représentation vectorielle des données
Recherche d'une situation source	Identification des voisins
Transfert des éléments inappariés de la source vers la cible	Inférence de la classe de la cible à partir de celles de ses voisins

TAB. 2.1 – Comparaison raisonnement par analogie - apprentissage paresseux

En revanche, les notions d'appariement et d'analogue dans le cas de l'apprentissage paresseux prennent un sens beaucoup plus restreint. Dans ce contexte, l'acception de l'analogie se confond avec celle de similarité. Une instance « analogue » est avant tout une instance proche. Ainsi, au questionnement formulé par Hall :

> Analogy presents a basic and challenging epistemological question: when are two representational descriptions, for some purpose alike?
>
> Hall (1989, p. 39)

les méthodes d'apprentissage paresseux répondent simplement : « quand elles sont proches ». Il suffit alors de munir l'espace d'entrée d'une mesure de similarité, et un appariement entre deux instances s'effectue si leur similarité dépasse un certain seuil [16] ; l'étape de transfert se résume à une simple recopie ($f(b) = f(a)$) , illustrée par la figure 2.16.

$$\begin{array}{ccccc} a & : & b & & a & : & b \\ & :: & & \rightarrow & & :: & \\ f(a) & : & ? & & f(a) & : & f(b) = f(a) \end{array}$$

FIG. 2.16 – Étape de transfert pour l'algorithme des *k*-ppv

Par conséquent, bien que les mécanismes généraux du raisonnement par analogie soient respectés, l'analogie est ici réduite au calcul d'une simple similarité et à un transfert par recopie ; nous parlerons d'*apprentissage par similarité*, version « dégénérée » de l'apprentissage par analogie. Les proportions impliquées dans l'étape de transfert n'exploitent aucune propriété structurelle des données. Or, comme nous l'avons remarqué avec Gentner, l'analogie repose avant tout sur une conservation de relations et de structures et ne peut se résumer à un calcul de similarité. Toutefois, cette simplification se justifie au regard de la nature des données usuellement manipulées par les apprentis paresseux. Ceux-ci n'ont pas pour vocation première la modélisation de capacités cognitives, mais doivent régulièrement fournir une réponse à des problèmes divers et variés tels que la classification automatique de textes, la détermination de la fonction d'une protéine, la classification d'images satellitaires, etc. Un aperçu de cette variété est par exemple observable à partir de l'*UCI Machine Learning Repository* (Hettich *et al.*, 1998). Dans

cf. Sec. 2.1.2, p. 18

16. Dans le cas des *k*-ppv, ce seuil est « dynamique » puisque dépendant de la configuration des exemples de la base d'apprentissage dans l'espace d'entrée au voisinage de l'exemple à analyser.

ces applications, les données sont souvent représentées sous forme vectorielle : un objet se présente sous la forme d'une combinaison d'attributs, nominaux, comme dans *ciel=dégagé*, ou continus comme dans *température=22°*. Nous dirons que ces représentations sont *plates* car ne rendant pas compte de la structure et des relations entre les éléments composant l'objet. Par exemple, le fait que l'attribut v_1 est toujours supérieur ou deux fois égal à l'attribut v_2 pour les éléments d'une certaine classe, n'apparaît pas de manière explicite dans la représentation vectorielle. Il existe néanmoins d'autres cadres d'apprentissage supervisé, tels que les Problèmes Multi-Instances (Dietterich *et al.*, 1997) ou la Programmation Logique Inductive (Lavrač & Džeroski, 1994), qui considèrent explicitement des représentations semi-relationnelles ou relationnelles. Toutefois, les apprentis paresseux proposés dans de tels cadres n'exploitent la structure des objets qu'au moyen, ici encore, d'une certaine mesure de similarité (Wang & Zucker, 2000 ; Ramon & Raedt, 1999 ; Horváth *et al.*, 2001 ; Emde & Wettschereck, 1996 ; Armengol & Plaza, 2001).

cf. Sec. 3.2, p. 51

Dans la suite, nous aurons l'occasion de discuter un modèle d'apprentissage qui considère une conception de l'analogie plus riche que la simple similarité, de façon à prendre en compte la nature structurée des représentations linguistiques : chaînes de symboles, arbres, graphes, automates, structures de traits, etc. Il s'agira de rétablir la propriété de systématicité dans un contexte d'apprentissage paresseux, à l'aide de l'exploitation de proportions analogiques. Ce modèle permettra en outre d'envisager des problèmes plus généraux que celui de la classification, impliquant des objets structurés à la fois dans l'espace d'entrée et dans l'espace de sortie.

2.3 Apprentissage du langage par analogie

2.3.1 Introduction

Nous avons étudié, dans les sections précédentes, la notion d'apprentissage par analogie, dont l'apprentissage par similarité est une version dégénérée. Cet apprentissage repose sur une inférence s'effectuant directement à l'aide d'« objets » mémorisés, exemples, cas, situations, etc., et s'oppose en ce sens aux modèles formulant une abstraction de connaissances. Cette inférence implique un appariement entre l'objet à analyser et les objets mémorisés, donnant lieu à un transfert fondé sur des proportions et des équations analogiques.

Dans un contexte linguistique, parler d'analogie est périlleux. En effet, une confusion sémantique entoure cette notion dont l'acception dépend des contextes et des locuteurs.

Par exemple, pour les grammairiens grecs et latins, l'analogie (qui signifie analogon=proportion) est synonyme de régularité et s'oppose à l'anomalie. Du processus d'apprentissage par analogie, seule la notion de proportion analogique est retenue. Puisqu'ils considèrent que ces proportions sont modélisables à l'aide de règles, l'analogie est dite « régulière ». Ici, les règles agissent localement, c'est-à-

dire relativement aux termes impliqués dans une proportion particulière.

À l'inverse, d'autres n'en conservent que son utilisation d'exemples, même si celle-ci conduit à une abstraction. Dans ce cadre, l'analogie peut être définie par la négative : elle regroupe tout ce qui ne relève pas de l'approche à base de règles ; en particulier, utiliser un réseau de neurones et une base d'exemples est une approche analogique [17]. C'est la définition utilisée dans l'opposition fréquente entre analogie et modèles à base de règles, notamment dans certaines études en psycholinguistique. Dans ce contexte, certaines des questions posées sont : est-ce que la connaissance linguistique est mémorisée sous forme de règles, sous forme d'exemples, y a-t-il des règles d'une part et des exceptions d'autre part, etc. ? Notons que les règles sont ici globales ; elles modélisent une connaissance générale sur un domaine.

Entreprendre une discussion sur les tenants et les aboutissants du débat autour de l'« analogie et ses acceptions en linguistique » dépasse clairement le cadre de notre travail. Dans la suite, nous fournissons simplement quelques exemples permettant d'éclairer succinctement deux ingrédients de l'apprentissage par analogie dans ce contexte, à savoir la notion de proportion analogique en linguistique, et l'apprentissage automatique de données linguistiques à partir d'exemples [18].

2.3.2 Linguistique et proportions analogiques

Les données linguistiques fournissent de nombreux exemples de proportions analogiques formelles, c'est-à-dire observables sur les formes. En particulier, ces dernières apparaissent clairement dans les paradigmes flexionnels et constructionnels de la morphologie des langues indo-européennes. Par exemple, on a, pour les formes fléchies des verbes en français,

« *marcher* est à *marchons* ce que *parler* est à *parlons* »,
« *prendre* est à *prenez* ce que *vendre* est à *venez* »,

pour les noms,

« *feuille* est à *feuilles* ce que *chat* est à *chats* »,
« *vitrail* est à *vitraux* ce que *bail* est à *baux* »,

et les adjectifs,

« *petit* est à *petite* ce que *grand* est à *grande* »,

etc.

17. Dans la section 2.2 concernant l'apprentissage paresseux, nous avons pourtant souligné qu'un réseau de neurones correspond à une abstraction des données : une fois les poids du réseau appris à l'aide d'exemples, ces derniers deviennent inutiles.

18. Ici, et dans le reste de cette section, apprentissage à partir d'exemples est bien synonyme d'apprentissage paresseux, tel que nous l'avons étudié dans la section 2.2.

De la même façon, la création de formes par construction implique régulièrement de telles proportions :

« *chanter* est à *chanteur* ce que *lutter* est à *lutteur* »,
« *généreux* est à *générosité* ce que *curieux* est à *curiosité* »,
« *agréable* est à *agréablement* ce que *arbitraire* est à *arbitrairement* »,

etc.

La présence de ces proportions analogiques n'a pas échappé aux linguistes, qui ont très vite pris la mesure de leur importance. Ainsi, les grammairiens alexandrins empruntent l'analogon des mathématiciens [19] pour établir des tableaux rendant compte des paradigmes de la morphologie flexionnelle grecque, ses déclinaisons et conjugaisons. De tels « tableaux de conjugaisons » sont, par ailleurs, utilisés encore aujourd'hui dans l'enseignement normatif de la langue française.

Au XIX^{e} siècle, les proportions analogiques apparaissent clairement chez Paul, Brugmann et Saussure, et sont décrites explicitement comme un modèle de production langagier. Cette production repose sur la *création analogique*, qui consiste à créer une nouvelle forme par résolution d'équation analogique. Ainsi :

> La plupart des formes […], nous les formons à l'aide de groupes, en mettant en rapport – de manière naturellement inconsciente – les grandeurs connues et en en déduisant la quatrième inconnue.
>
> Brugmann

Cette création implique donc la résolution d'une équation analogique « *a* est à *b* ce que *c* est à *?* », où '?' est la *quatrième inconnue*. Pour Saussure, l'analogie offre également un mécanisme de réparation aux transformations dues au changements phonétiques.

> L'analogie suppose un modèle et son imitation régulière. *Une forme analogique est une forme faite à l'image d'une ou plusieurs autres d'après une règle déterminée.*
>
> Ainsi, le nominatif latin *honor* est analogique. On a dit d'abord *honôs : honôsem*, puis par rotacisation de l'*s honôs : honôrem*. Le radical avait dès lors une double forme ; cette dualité a été éliminée par la forme nouvelle *honor*, créée sur le modèle de *ôrâtor : ôrâtôrem*, etc., par un procédé que nous […] ramenons au calcul de la quatrième proportionnelle :
>
> *ôrâtôrem : ôrâtor = honôrem : x.*
> *x = honor.*
>
> […] Pour contrebalancer l'action diversifiante du changement phonétique (*honôs : honôrem*), l'analogie a de nouveau unifié les formes et rétabli la régularité (*honor : honôrem*).
>
> de Saussure (1916, p. 221-222)

L'analogie implique, ici aussi, la résolution d'une équation analogique, dénommée *calcul de la quatrième proportionnelle*.

19. Rappelons que l'analogie est présentée par Aristote comme un rapport de proportions (analogon) : « j'entends par rapport d'analogie tous les cas où le second terme est au premier comme le quatrième au troisième, car le poète emploiera le quatrième au lieu du second et le second au lieu du quatrième ».

Bloomfield (1970) invoque des principes similaires dans le cadre de la syntaxe. Ces principes seront fortement remis en cause par Chomsky (1957), donnant lieu au mouvement génératiste. Toutefois, la plupart des arguments opposés à l'approche analogique reposent sur une vision de celle-ci réduite à la simple similarité ou ressemblance de surface ; le combat mené contre cette version appauvrie est à l'évidence inégal [20]. Dans ce contexte, la confusion sémantique évoquée précédemment contribue à l'entretien du flou général autour de la notion d'analogie. En réponse aux arguments des génératives, et de manière à rééquilibrer le combat, une « réhabilitation de l'analogie » sera avancée d'abord par Itkonen & Haukioja (1997), puis par Lepage (2001) et Lavie (2003). Soulignons qu'on peut trouver chez Lepage (2003) une analyse détaillée de l'histoire de la notion d'analogie en linguistique et ailleurs. Nous reviendrons sur les proportions analogiques dans les données linguistiques dans la section 3.4, après avoir introduit le modèle d'apprentissage les exploitant.

2.3.3 TAL et apprentissage à partir d'exemples

Parallèlement aux considérations théoriques issues de l'étude linguistique, un certain nombre d'observations peuvent être effectuées au regard de la situation actuelle dans le domaine du Traitement Automatique des Langues. Celui-ci a vu se développer, depuis les années 1950, des systèmes dits à base de règles, reposant fortement sur les modèles génératifs évoqués plus haut. Dans ces systèmes, la connaissance linguistique s'exprime par des représentations symboliques d'entités linguistiques et des règles permettant d'effectuer des traitements sur ces représentations. Les règles peuvent par exemple prendre la forme de règles de production ou de réécriture pour le traitement syntaxique, ou de règles d'inférence pour le traitement sémantique. La construction de telles bases de règles, si l'on attend d'elles qu'elles soient de bonne qualité, est très coûteuse. D'une part, les experts ne sont pas toujours en mesure d'expliciter leur expertise [21]. D'autre part, la connaissance accumulée sur un domaine ne se transporte pas toujours simplement sur un autre domaine. Chaque nouveau besoin implique potentiellement de lourdes adaptations et un coût associé élevé, rendant ces approches in fine peu rentables.

De manière à s'affranchir de ces limitations, des méthodes ont cherché à effectuer une exploitation de corpus de données linguistiques annotées. De même qu'en apprentissage automatique, on distingue les méthodes induisant un modèle (généralisation) et les méthodes inférant directement à partir d'exemples (apprentissage paresseux). L'approche inductive a principalement conduit à la proposition de modèles probabilistes ; cette démarche a connu un fort essor durant la dernière décennie et a fourni des réponses souvent efficaces à des problèmes variés.

20. Lavie (2003, p. 36) souligne par ailleurs que la mise en garde contre une vision de l'analogie comme une simple similarité de surface est déjà formulée par Saussure : « La combinaison *ôrâtôrem : ôrâtor :: honôrem :x → x = honor* n'aurait aucune raison d'être si l'esprit n'associait pas par leur sens les formes qui la composent. »

21. Selon Chomsky, tout locuteur est un « expert » ; cela n'est toutefois pas suffisant à rendre explicite cette expertise.

Dans la suite, nous présentons les caractéristiques principales de ces trois familles d'approches.

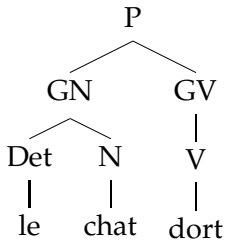
Les systèmes à base de règles

Les systèmes à base de règles travaillent essentiellement sur des représentations symboliques d'entités linguistiques et des règles permettent d'effectuer des traitements sur ces entités. Prenons l'exemple de la transcription orthographique/phonétique en français. Dans cette tâche, il s'agit de proposer une ou plusieurs séquences de symboles phonétiques correspondant à une séquence de symboles graphiques fournie en entrée. Par exemple, pour la séquence *apaise*, la sortie attendue est apɛz. Pour résoudre un tel problème, un système à base de règles exploite des règles exprimées selon un certain formalisme. Par exemple, on peut considérer des règles du type [22] :

- [*a*] → a
- [*a*]*i* → ɛ
- %V [*s*] %V → z

signifiant respectivement que le symbole *a* est par défaut transcrit en le phonème a, que suivi d'un symbole *i*, il se prononce ɛ, et qu'un *s* entre deux voyelles se prononce z. Un tel ensemble de règles, accompagné d'un moteur capable de les traiter, permet de transcrire une séquence orthographique en une séquence phonétique, et d'associer à l'entrée *apaise* la sortie apɛz.

Prenons un autre exemple, celui de l'analyse syntaxique. Dans ce contexte, il s'agit d'associer à la chaîne *le chat dort* l'arbre syntaxique



exprimant l'organisation de la phrase en constituants syntagmatiques (P = Phrase, GN = Groupe Nominal, GV = Groupe Verbal) et lexicaux (Det = Déterminant, N = Nom, V = Verbe). Pour effectuer cette analyse, on peut considérer une grammaire hors-contexte, exprimée par l'ensemble des règles de réécriture suivant :

- P → GN GV
- GN → Det N
- GV → V
- Det → le
- N → chat
- V → dort

22. Ce formalisme est celui utilisé par le logiciel de transcription orthographique/phonétique DIMI (Régis-Gianas & Yvon, 2004).

Celles-ci expriment de quelles façon les constituants se réécrivent ; par exemple, on peut réécrire un groupe nominal (GN) en un déterminant (Det) suivi d'un nom (N). Les séquences de mots que l'on obtient par réécritures successives à partir du symbole initial 'P' (phrase) constituent le langage *généré* par la grammaire. De cette grammaire il est possible de dériver un analyseur syntaxique capable de traiter n'importe quelle séquence de mots. Si celle-ci appartient au langage généré par la grammaire, alors un tel analyseur est en mesure de fournir la trace de l'analyse, sous la forme d'un arbre (dit de dérivation) identique à l'exemple donné plus haut (cf. Grune & Jacobs (1990) sur de telles méthodes d'analyse). Gazdar & Mellish (1989) fournissent une vue d'ensemble de ce type d'approches.

Les approches probabilistes

Les approches probabilistes (paramétriques) reposent sur l'hypothèse d'un modèle probabiliste prenant la forme d'une distribution (inconnue) $\mathbb{P}$ (Charniak, 1996 ; Manning & Schütze, 1999). En général, on considère une famille de distributions, dépendant d'un paramètre θ ; nous dirons que $(\mathbb{P}_\theta, \theta \in \Theta)$ est un *modèle paramétrique*. Le paramètre θ est à estimer à partir d'un corpus de données (dit corpus d'apprentissage). Dans le cas d'une tâche d'analyse, dans laquelle il s'agit d'associer une sortie y à une entrée x, ce corpus d'apprentissage peut être constitué de couples (x_i, y_i), associant un objet et son analyse. Par exemple, pour une tâche de transcription orthographique/phonétique, x_i est une séquence de symboles graphiques (telle que *apaise*), et y_i une séquence de phonèmes (telle que $^{\text{ap}\varepsilon\text{z}}$). Une fois le paramètre θ estimé, la distribution permet d'assigner une probabilité aux nouvelles données rencontrées. En particulier, pour une nouvelle donnée à analyser x, elle permet de calculer $\mathbb{P}(y|x)$ pour tout y. L'analyse $h(x)$ alors proposée pour x correspond au y maximisant la quantité $\mathbb{P}(y|x)$:

$$h(x) = \underset{y}{\text{argmax}}\, \mathbb{P}(y|x).$$

On retrouve ici une formulation présentant des similitudes avec la tâche de classification automatique, évoquée dans le cadre de l'apprentissage automatique. Dans cette tâche, nous disposons également d'un ensemble de données composant une base d'apprentissage $BA = \{(x_i, y_i), \ldots (x_n, y_n)\}$. Chaque exemple (x_i, y_i), issu d'une distribution inconnue $\mathbb{P}$, est un couple représentant un objet et sa classe associée. À partir de cette base, la tâche d'apprentissage consiste à construire un apprenti g assignant à tout x une valeur $g(x)$ prédisant sa classe. On peut, ici aussi, construire g à partir de $\mathbb{P}$, selon $g(x) = \text{argmax}_y\, \mathbb{P}(y|x)$.

cf. Sec. 2.2.1, p. 23

La différence principale entre ces problèmes réside dans la nature des données manipulées. Dans un problème de classification, un exemple (x, y) correspond habituellement à un couple constitué d'un vecteur d'attributs x associé à une classe y. Les applications en TAL impliquent généralement des données séquentielles (séquences de graphèmes, de phonèmes, de mots, d'étiquettes morpho-syntaxiques, etc.), aussi bien en entrée qu'en sortie. Ainsi, pour une tâche de transcription orthographique/phonétique, x est une séquence de symboles graphiques $x_1 \ldots x_m$ et

y une séquence de phonèmes $y_1 \dots y_l$. De nombreux problèmes étudiés en traitement automatique des langues peuvent être formulés dans un tel cadre : la reconnaissance de la parole (séquence phonétique → séquence orthographique), l'étiquetage morpho-syntaxique (séquence de mots → séquence d'étiquettes morpho-syntaxiques), la traduction automatique (séquence de mots dans une langue a → séquence de mots dans une langue b), etc. De telles tâches ne s'expriment pas naturellement dans le cadre de la classification ; des outils plus spécifiques ont alors été développés. Citons les modèles de Markov cachés (*Hidden Markov Model* en anglais, HMM) (Rabiner, 1989), les HMM entrées-sorties (Bengio & Frasconi, 1996), les HMM à maximum d'entropie (McCallum *et al.*, 2000), et les champs aléatoires conditionnels (Lafferty *et al.*, 2001). Dietterich (2002) expose avec clarté les liens entretenus par les problématiques de l'apprentissage de données séquentielles et la classification.

cf. Sec. 3.1.2, p. 48

Pour que ces modèles donnent lieu à des méthodes efficaces, il convient d'effectuer un certain nombre d'hypothèses. Effectuer trop d'hypothèses conduit à un pouvoir d'expression réduit, échouant à rendre compte des données avec précision. Réciproquement, les modèles trop souples impliquent un nombre élevé de paramètres qu'il est difficile d'estimer efficacement et correctement. Choisir un modèle ainsi que son dimensionnement se révèle être le problème majeur de telles approches. Notons également que l'ajout de connaissances linguistiques disponibles n'est pas naturel dans un tel cadre.

Apprentissage de données linguistiques à partir d'exemples

Les deux familles d'approches présentées ci-dessus reposent sur une généralisation des données : ensemble de règles pour l'une, paramètres d'un modèle probabiliste pour l'autre. Un apprentissage à partir d'exemples remet en cause ce principe d'abstraction. La connaissance linguistique reste alors implicitement représentée dans les exemples constituant le corpus accumulé : ils contiennent en puissance toute la connaissance requise au traitement de données nouvelles.

Selon cette approche, pour prononcer *marchons*, nul besoin de règles, nul besoin de calculer $\mathbb{P}(y|marchons)$; savoir prononcer *marcher*, *parler* et *parlons* suffit. Nous présentons, dans la suite, trois exemples de systèmes suivant cette approche.

L'équipe de W. Daelemans Les membres de l'équipe de Walter Daelemans à l'Université de Tilburg ont appliqué l'apprentissage à partir d'exemples (Daelemans & van den Bosch, 2005) dans de nombreux contextes applicatifs : analyse morphologique (van den Bosch & Daelemans, 1999), prononciation (Daelemans & van den Bosch, 2001), résolution de l'attachement des syntagmes prépositionnels (Zavrel *et al.*, 1997), désambiguïsation lexicale (Veenstra *et al.*, 2000) et étiquetage morpho-syntaxique (Daelemans *et al.*, 1996).

Pour prononcer *marchons*, chaque lettre est considérée indépendamment : la prononciation de *r* s'obtient en cherchant dans la base d'apprentissage tous les

exemples où *r* apparaît dans un contexte semblable, par exemple *parchemin* (*r* est ici aussi précédé de *a* et suivi de *ch*), ou simplement *marcher*. La même procédure s'applique à toutes les lettres, permettant la prononciation de la forme entière.

Dans ce modèle, l'inférence s'effectue donc directement à partir des exemples, sans construction d'abstraction. En réalité, leur approche consiste à transformer tout problème rencontré en un problème de classification, puis à traiter ce problème de classification par une méthode d'apprentissage paresseux (*k*-ppv ou variante). L'apprentissage n'est donc pas effectué directement sur les représentations linguistiques, mais sur des reformulations de celles-ci. Nous aurons l'occasion de décrire plus longuement ce modèle dans la suite.

cf. Sec. 3.1.2, p. 45

Skousen La méthode d'apprentissage par analogie proposée par Skousen (1989) implique, elle aussi, la réexpression d'un problème sous la forme d'une tâche de classification. Ici, l'algorithme utilisé pour la résolution de cette tâche n'est pas les *k*-ppv, mais un algorithme spécifique.

Pour prononcer le *r* de *marchons*, on ne va pas chercher les voisins de *arc* (*r* + contextes immédiats gauche et droit), mais des « supra-contextes » de celui-ci. Les supra-contextes de *arc* sont formés à partir des exemples de la base d'apprentissage contenant les contextes *arc*, *ar*, *rc*, etc. Ceux-ci représentent les contextes plus généraux que *arc*. Si les exemples formant un supra-contexte sont d'accord sur la prononciation à proposer (i.e. ils appartiennent à la même classe), alors le supra-contexte est dit homogène [23]. La prononciation pour *r* est ensuite calculée à partir des supra-contextes homogènes de *arc* (par exemple en prenant la prononciation la plus fréquemment proposée). Daelemans *et al.* (1997*a*) compare cette approche et l'apprentissage paresseux traditionnel.

Traduction Automatique à partir d'exemples Dans le domaine de la Traduction Automatique (TA), l'apprentissage à partir d'exemples s'est développé à partir de considérations spécifiques au domaine.

Citons tout d'abord le cas des *mémoires de traduction* : celles-ci sont constituées de phrases dans une langue *A*, accompagnées de leur traduction dans une langue *B* (Arthern, 1978 ; Kay, 1980). Une telle mémoire de traduction correspond à ce que nous avons jusqu'à maintenant appelé base d'apprentissage. Dans l'approche traditionnelle à base de mémoire de traduction, l'apprentissage effectué est un simple apprentissage par cœur : si la phrase à traduire est contenue dans la mémoire, alors la phrase correspondante dans la langue cible est proposée ; sinon, aucune solution n'est générée. À première vue, un tel apprentissage par cœur peut sembler peu performant. En réalité, dans le contexte de la traduction automatique [24], ce genre d'approche peut être utile dans de nombreuses applications. Ainsi, on peut raisonnablement penser qu'une grande partie des phrases de la version *x* d'un manuel

23. Bien entendu, plus le contexte est général, moins il a de chance d'être homogène.

24. Pour être plus précis, il s'agit ici davantage d'aide à la traduction que de traduction véritablement automatique.

est également contenue dans la version $x - 1$ du même manuel : si cette dernière est déjà traduite, utiliser une mémoire de traduction permet de faciliter grandement la tâche. De même, certains documents administratifs ou législatifs peuvent contenir des constructions relativement redondantes et réexploitables.

Un certain nombre d'approches ont cherché à traduire des phrases non rencontrées à l'aide de bases de phrases traduites (Nagao, 1984 ; Sato & Nagao, 1990 ; Sumita *et al.*, 1990). Nagao (1984) propose ainsi de traduire la phrase : *he eats potatoes* à l'aide de la phrase *a man eats vegetables* et de sa traduction en japonais *hito-wa yasai-wo taberu*. Après avoir identifié un appariement entre *a man* et *he*, *vegetables* et *potatoes*, et en utilisant des connaissances sur la langue cible (*he* se traduit *kare*, *potatoes* se traduit *jagaimo*), la traduction proposée est *kara-wa jagaimo-wo taberu* (cf. figure 2.17).

a man eats vegetables : *he eats potatoes* :: *hito-wa yasai-wo taberu* :?
? → *kare-wa jagaimo-wo taberu*

FIG. 2.17 – Traduction de Nagao

Les questions qui se posent alors à de telles méthodes sont caractéristiques de l'apprentissage par analogie : quels sont les exemples utiles à l'inférence, comment effectuer l'appariement et le transfert permettant cette inférence, sur quoi repose l'adaptation de connaissances, etc. ? Des réponses spécifiques ont alors été proposées, constituant le domaine de la traduction automatique à partir d'exemples ; pour une vue d'ensemble de ces techniques, voir Somers (1999) et Carl & Way (2003). Voir également Somers (2001) pour une comparaison entre la traduction à partir d'exemples et le raisonnement à partir de cas. Notons finalement que si les modèles de Daelemans et Skousen se rapprochent tous les deux de l'apprentissage paresseux traditionnel, la traduction automatique à partir d'exemples s'appuie souvent implicitement sur des proportions analogiques.

2.4 Conclusion

Quel que soit son champ d'application, l'apprentissage par analogie repose sur une inférence s'effectuant directement à l'aide d'« objets » mémorisés, exemples, cas, situations, entités linguistiques, etc. Il s'oppose en ce sens à la formulation d'une abstraction de connaissances, que ce soit sous forme de règles ou de modèle explicitable. Selon les objectifs et les applications visés, cette approche peut prendre différentes formes. À l'une des extrémités du spectre, on trouve les modèles de raisonnement issus des sciences cognitives. Ces modèles impliquent des situations complexes et structurées, mais peu nombreuses. L'analogie prend alors la forme d'un appariement complexe entre structures relationnelles. À l'autre extrémité, on trouve les apprentis paresseux, traitant de larges bases de données faiblement structurées. Dans ce contexte, l'analogie prend une forme dégénérée, réduite à une simple mesure de similarité.

L'approche analogique possède par ailleurs un certain nombre de fondements théoriques dans l'étude linguistique, nous incitant à considérer une conception riche de l'analogie, c'est-à-dire donnant un sens à la notion de proportion analogique, et capable de tenir compte de la nature structurée des représentations linguistiques. Nous verrons, dans la suite, comment intégrer ces considérations dans un cadre d'apprentissage automatique, de manière à exploiter les bases de données disponibles tout en respectant le sens originel de l'analogie, c'est-à-dire réconcilier l'apprentissage paresseux et l'exploitation de proportions.

CHAPITRE

3

Exploitation de proportions analogiques pour l'Apprentissage Automatique du Langage Naturel

I began to look about and write down the elements with their atomic weights and typical properties, analogous elements and like atomic weights on separate cards, and this soon convinced me that the properties of elements are in periodic dependence upon their atomic weights.

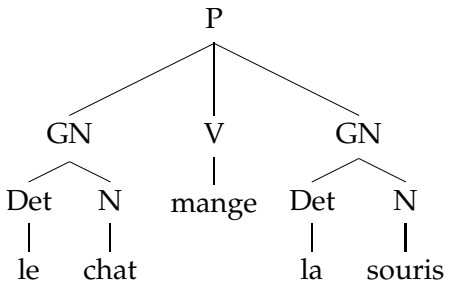
— Mendeleev, *Principles of Chemistry, vol. II, 1905.*

– « c'est bien mais il faudrait rajouter de l'aujourd'huité »,
tu sais ce que je lui ai répondu ?
« Et la demainitude aussi ? »
– Moi j'ai un chef de groupe qui me parle tout le temps de la gustativité ! Il ne connaît pas le mot goût !

— Frédéric Beigbeder, *99F*

Sommaire du chapitre

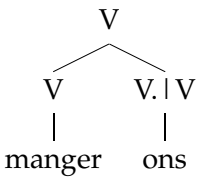
DANS le domaine du Traitement Automatique des Langues (TAL), de nombreuses tâches s'expriment comme le passage d'un niveau de représentation à un autre. Étudier le passage d'un objet linguistique x d'un niveau de représentation A à un autre niveau B consiste à analyser x selon B à partir de sa représentation selon A. Prenons un exemple. L'analyse syntaxique d'une phrase p peut correspondre au passage de la représentation de p sous la forme de séquence de symboles graphiques « *le chat mange la souris* », à sa représentation sous la forme d'une structure hiérarchique arborescente, comme



De même, l'analyse morphologique d'un mot m peut correspondre au passage de la séquence « *mangerons* » à une combinaison d'un lemme et d'un ensemble d'attributs morpho-syntaxiques, comme

$$\begin{bmatrix} \textit{Lemme} & : \textit{manger} \\ \textit{Personne} & : 1^{\text{re}} \\ \textit{Nombre} & : \textit{pluriel} \\ \textit{Temps} & : \textit{présent de l'indicatif} \end{bmatrix}$$

ou à une structure hiérarchique mettant en évidence ses constituants morphémiques, comme



De nombreuses tâches en TAL peuvent s'exprimer selon un passage similaire : analyse morphologique, analyse syntaxique, prononciation, transcription phonétique/orthographique, traduction automatique, etc.

Un des objectifs de notre travail est d'**étudier comment effectuer une telle analyse de manière automatique, en exploitant des données linguistiques déjà analysées, et en suivant les principes de l'apprentissage par analogie**. Cette approche, décrite et motivée dans le chapitre précédent, invite en particulier à donner un sens à la notion de proportion analogique. Une proposition dans ce sens sera faite dans le chapitre 4. Dans le présent chapitre, nous formalisons les tâches d'analyse de changement de niveau de représentation en les resituant dans un cadre général d'apprentissage automatique supervisé.

Les deux problèmes les plus étudiés en apprentissage automatique supervisé sont ceux de la classification et de la régression [1]. Les tâches que l'on se propose de traiter ne s'adaptent pas naturellement à ces deux contextes. En particulier, nous observons dans la section 3.1 que les méthodes d'apprentissage recherchées doivent être capables de construire des apprentis présentant des propriétés de symétrie relativement aux espaces d'entrée et de sortie. Cette propriété entraîne d'autres pré-requis ; les apprentis doivent être en mesure de : (i) traiter des objets structurés aussi bien en entrée qu'en sortie du système, (ii) pouvoir proposer des solutions non contenues dans la base d'apprentissage [2]. Nous présentons dans la section 3.2 une méthode d'apprentissage reposant sur la notion de proportion analogique, et répondant aux besoins spécifiques des tâches ciblées. Nous étudions ensuite les caractéristiques de cette méthode à l'aide de la notion d'extension analogique (section 3.3) et discutons sa pertinence dans un contexte de traitement automatique de données linguistiques (section 3.4).

3.1 Apprentissage automatique de changement de niveau de représentation

3.1.1 Introduction

Les méthodes issues de l'apprentissage automatique fournissent un certain nombre de réponses au problème de la classification. Dans ce problème, évoqué précédemment, il s'agit d'affecter automatiquement à un objet x une certaine classe y, en s'appuyant sur une base d'apprentissage BA constituée d'objets x_i dont la classe y_i est connue, i.e. $BA = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, avec $(x_i, y_i) \in X \times Y$, X étant l'espace d'entrée et Y l'espace de sortie. Dans cette tâche de classification, les objets x_i sont habituellement représentés par des vecteurs d'attributs (numériques ou symboliques) et l'espace de sortie Y est l'ensemble fini des classes possibles. Les tâches (d'analyse) de changement de niveau de représentation, telles que celles exposées ci-dessus, impliquent généralement des objets structurés (séquences, arbres) à la fois dans l'espace d'entrée et de sortie et ne s'expriment pas naturellement dans un cadre de classification. En parlant d'un cadre d'apprentissage dans lequel les sorties peuvent être structurées, Weston *et al.* (2003) remarquent :

cf. Sec. 2.2.1, p. 23

> The framework we attempt to address is rather general. Few algorithms have been constructed which can work in such a domain - in fact the only algorithm that we are aware of is *k*-nearest neighbors. Most algorithms have focussed on the pattern recognition and regression problems and cannot deal with more general outputs.
>
> Weston *et al.* (2003)

1. Dans un problème de régression, l'espace de sortie Y est l'ensemble des réels $\mathbb{R}$.
2. Cette deuxième contrainte est plus forte qu'il n'y paraît. Elle n'exprime pas qu'il faille savoir analyser une entrée absente de la base d'apprentissage : c'est l'essence même de l'apprentissage automatique ; elle exprime le fait que la sortie attendue pour une entrée donnée peut n'avoir encore jamais été rencontrée. Ce point sera détaillé dans la suite.

Nous étudions dans la suite les particularités des tâches de changement de niveau de représentation vues comme des problèmes d'apprentissage automatique.

Tout d'abord, les espaces d'entrée et de sortie présentent une certaine symétrie. En effet, chaque niveau de représentation contient une « information » comparable relativement à l'objet linguistique sous-jacent. Ainsi, les représentations *mangerons* et *manger+1re+pluriel+futur_indicatif* correspondent à deux visions d'une même entité linguistique, et passer de l'une à l'autre des représentations ne conduit pas nécessairement à une très grande perte d'information. À l'inverse, dans le contexte d'une tâche de classification, la classe d'un objet ne permet en aucun cas de le reconstruire. Dans ce cas, l'espace d'entrée, potentiellement riche, et l'espace de sortie, composé d'un ensemble fini de classes, ne sont nullement interchangeables, et la projection effectuée de l'espace d'entrée sur l'espace de sortie lors de la classification est un processus irréversible.

La non symétrie des problèmes de classification s'observe dans l'étude des espaces d'entrée et de sortie. Pour une base d'apprentissage

$$BA = \{(x_1, y_1), \ldots, (x_n, y_n)\},$$

posons $BA_X = \cup_i \{x_i\}$ et $BA_Y = \cup_i \{y_i\}$.

L'objectif de la classification automatique est de pouvoir affecter à un $x \in X$ non connu une certaine classe $y \in Y$. Si tous les x visés sont dans BA_X, la tâche est trivialement résolue (par un apprentissage par cœur) et le problème inexistant. Les cas intéressants correspondent donc à $BA_X \subsetneq X$. En revanche, l'ensemble des classes contenues dans la base d'apprentissage se confond généralement avec l'espace de sortie, et on peut attendre d'un apprenti g que son image couvre également cet espace, ce qui s'exprime par :

$$BA_Y = g(X) = Y.$$

La plupart des méthodes d'apprentissage font cette hypothèse, totalement justifiée quand l'ensemble des classes est fini et déterminé a priori.

Cependant, dans les cas qui nous intéressent, l'entrée x et la sortie y correspondent à deux visages d'une même réalité linguistique. Ainsi, si x est absent de la base d'apprentissage ($x \notin BA_X$), il y a de forts risques qu'il en soit de même pour y ($y \notin BA_Y$). Par exemple, dans le cas de l'apprentissage d'analyses morphologiques, si la séquence *mangerons* est absente de la base d'apprentissage, la sortie attendue *manger+1re+pluriel+futur_indicatif* sera elle aussi absente de la base d'apprentissage. Nous avons par conséquent dans le cas général :

$$BA_Y \subsetneq g(X) \subsetneq Y.$$

Les méthodes ne proposant pas de solution en dehors de BA_Y ($g(X) = BA_Y$) ont donc toutes les chances d'échouer face à de telles tâches. Pour espérer obtenir des résultats satisfaisants, le modèle d'apprentissage doit prendre en compte cette particularité, et autoriser $BA_Y \subsetneq g(X)$.

Soulignons enfin que les tâches que l'on se propose d'étudier sont confrontées naturellement à la présence d'ambiguïtés. En effet, il n'y a pas de correspondance biunivoque entre une forme graphique et une séquence de phonèmes, une séquence de phonèmes et un sens, une séquence de mots et une analyse syntaxique, etc. ; la forme graphique *marche* peut désigner un verbe ou un nom, *ananas* se prononce anana ou ananas. Il est donc légitime d'attendre d'un système qu'il soit capable de prendre en compte ces ambiguïtés, en proposant éventuellement plusieurs solutions pour une entrée donnée.

3.1.2 Quelques réponses

Pour s'adapter à ces problèmes particuliers, plusieurs types de démarches ont été envisagées.

Classer à tout prix Une réponse consiste à essayer de reformuler tous les problèmes rencontrés comme des problèmes de classification. L'intérêt d'une telle démarche est clair : une fois la reformulation effectuée, toutes les méthodes d'apprentissage répondant au problème de la classification sont théoriquement applicables (*k*-ppv, SVM, arbres de décisions, réseaux de neurones, etc.). Cette reformulation a pour objectif de convertir une base d'apprentissage

$$BA = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subset X \times Y$$

en une base

$$BA' = \{(x'_1, y'_1), \ldots, (x'_m, y'_m)\} \subset X' \times Y',$$

où Y' est un ensemble fini (réduit) de classes. Une telle conversion dépend de la nature des objets considérés. Dans la suite, nous décrivons une méthode, dite de *fenêtrage*, qui effectue la transformation en question dans le cas où les entrées et les sorties sont des séquences.

Dans cette méthode, chaque couple $(x_i, y_i) \in BA$ est converti en un ensemble de couples $\{(x_{i1}, y_{i1}), \ldots (x_{ik}, y_{ik})\}$. Pour effectuer cette conversion, on aligne tout d'abord l'entrée et la sortie de façon à ce qu'elles soient représentées par des séquences de même taille[3]. Par exemple, pour la forme *marcher* et la prononciation associée, on peut obtenir l'alignement illustré sur le tableau 3.1. Le symbole '_' désigne l'absence de phonème.

Ensuite, pour chaque position i de cet alignement, on considère une fenêtre de taille k, de manière à prendre en compte les contextes gauche et droit du symbole graphique correspondant à la position i dans la forme graphique. Cette fenêtre est centrée sur le symbole graphique en question. Ce procédé est illustré sur la figure 3.2 avec $k = 5$. Le symbole '-' est utilisé de façon à pouvoir représenter artificiellement le contexte gauche (resp. droit) des premières (resp. dernières) lettres d'un mot.

3. Cette phase n'est pas nécessairement triviale ; résoudre le problème de l'alignement, c'est répondre en partie au problème de l'apprentissage (cf. section 5.2.3).

position	graphème	phonème
1	m	m
2	a	a
3	r	r
4	c	ʃ
5	h	_
6	e	e
7	r	_

TAB. 3.1 – Alignement entre la forme *marcher* et sa prononciation

position	fenêtre
1	- - **m** a r
2	- m **a** r c
3	m a **r** c h
4	a r **c** h e
5	r c **h** e r
6	c h **e** r -
7	h e **r** - -

TAB. 3.2 – Fenêtrage de *marcher*

Ensuite, on associe à chacune de ces fenêtres le phonème correspondant dans l'alignement (cf. figure 3.3). Ainsi, l'exemple $(\textit{marcher}, \mathrm{marʃe})$ est remplacé par l'ensemble d'exemples : $(\langle -, -, m, a, r\rangle, \mathrm{m})$, $(\langle -, m, a, r, c\rangle, \mathrm{a})$, $(\langle m, a, r, c, h\rangle, \mathrm{r})$, etc. Chaque exemple dans la nouvelle représentation est composé d'un vecteur d'attributs symboliques de taille fixe k, tel que $\langle -, -, m, a, r\rangle$, et d'une classe, telle que m. L'espace d'entrée est désormais E^k, où E est l'ensemble des valeurs que peuvent prendre les attributs (ici les graphèmes) ; l'espace de sortie Y correspond à l'ensemble des classes possibles (ici les phonèmes). Cette nouvelle tâche d'apprentissage peut ensuite être traitée par la plupart des méthodes d'apprentissage supervisé courantes. Notons que, dans ce cadre, on effectue une hypothèse d'indépendance des sorties ; la décision prise sur une classe pour une position donnée i n'influe pas sur celles à prendre pour les positions $j \neq i$.

cf. Sec. 2.3.3, p. 36

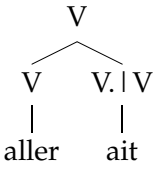
Cette démarche est celle adoptée, par exemple, par l'équipe de Walter Daelemans (cf. Daelemans & van den Bosch (2005)). Des transformations telles que la méthode de fenêtrage présentée ci-dessus sont opérées pour traiter des tâches comme l'analyse morphologique (van den Bosch & Daelemans, 1999), la prononciation (Daelemans & van den Bosch, 2001), la résolution de l'attachement des syntagmes prépositionnels (Zavrel *et al.*, 1997), la désambiguïsation lexicale (Veenstra *et al.*, 2000) et l'étiquetage morpho-syntaxique (Daelemans *et al.*, 1996). Le méca-

position	contexte gauche		graphème ciblé	contexte droit		phonème (= classe)
1	-	-	**m**	a	r	m
2	-	m	**a**	r	c	a
3	m	a	**r**	c	h	r
4	a	r	**c**	h	e	ʃ
5	r	c	**h**	e	r	_
6	c	h	**e**	r	-	e
7	h	e	**r**	-	-	_

TAB. 3.3 – Exemples de la nouvelle tâche de classification

nisme d'apprentissage utilisé, implanté dans le logiciel TIMBL (Daelemans *et al.*, 2004), repose sur un apprentissage paresseux dérivant de l'algorithme des *k*-ppv.

Dans cette approche, l'hypothèse la plus forte réside dans la possibilité d'effectuer la reformulation du problème, et ceci sans perte d'information [4]. Concernant ce second point, notons que la méthode de fenêtrage évoquée ci-dessus ne permet pas de capturer des dépendances entre des lettres éloignées dans un mot ; ces dépendances n'apparaissent plus dans le problème reformulé si la taille de la fenêtre est trop petite. Par exemple, pour savoir que *en* se prononce ɑ̃ dans *scientifique* et ne se prononce pas dans *apprécient*, la fenêtre *cient* n'est pas suffisante. Soulignons, sur le premier point, que si la transformation opérée dans le cas des séquences n'apparaît pas naturelle, elle l'est encore moins lorsque les objets manipulés sont davantage structurés. En effet, que signifie un alignement entre *irait* et



,

exemple pour lequel une allomorphie est observée ?

Cette démarche, aussi efficace soit-elle, présente ainsi des limites lorsque l'on veut traiter des objets complexes tels que les arbres. Ce comportement sera mis en évidence expérimentalement dans la suite. Il est par ailleurs notable que parmi l'ensemble des problèmes traités par Daelemans et ses coauteurs avec TIMBL, aucun n'implique de telles structures. En outre, le choix de la méthode de reformulation est capital ; des reformulations différentes peuvent conduire à des résultats sensiblement différents. Or, aucun type de reformulation ne s'impose a priori.

cf. Sec. 5.4.4, p. 144

4. Pour être complet, nous devrions ajouter : et sans « tricher ». En effet, le choix d'une reformulation particulière est susceptible d'introduire un biais notable, conduisant non pas à une perte mais à un ajout d'information. Une approche honnête se doit de justifier l'introduction d'un tel biais. Nous reviendrons sur ce point dans la section 5.2.3.

Réordonner Une autre approche consiste à construire un système capable d'ordonner les solutions qu'on lui propose. Un tel « ordonnanceur » doit être utilisé conjointement à un mécanisme proposant des solutions pour une entrée donnée. Cette approche est celle suivie par exemple par Collins & Duffy (2001) ; Collins & Koo (2005) ; Charniak & Johnson (2005) ; Carreras *et al.* (2005) ; Shen & Joshi (2005).

Leur méthode apprend une fonction F associant à toute sortie y une valeur $F(y)$. Plus cette valeur est élevée, meilleure est la solution. Nous n'exposons pas ici les détails des méthodes d'apprentissage utilisées. Cette fonction est ensuite couplée à un mécanisme capable de proposer un ensemble de solutions $Gen(x)$ pour une entrée x. Dans l'application qu'ils étudient, à savoir l'analyse syntaxique, l'entrée x est une chaîne et la sortie y un arbre ; le mécanisme générateur de propositions utilisé repose sur une grammaire hors-contexte probabilisée. Celle-ci propose pour une entrée x, un ensemble d'arbres $Gen(x)$. À l'aide de ces deux ingrédients, le système prédit alors $f(x)$ selon $f(x) = argmax_{y \in Gen(x)} F(y)$.

La limitation principale du modèle réside dans sa dépendance à un mécanisme producteur de propositions. En effet, pour que le système fonctionne, les propositions contenues dans $Gen(x)$ doivent être « bonnes », c'est-à-dire assez proches de la sortie attendue. En particulier, cette dernière doit faire partie de l'ensemble des propositions si l'on veut pouvoir l'inférer. Or, proposer de bonnes solutions, c'est déjà résoudre en partie le problème de l'apprentissage.

Notons enfin que le problème traité par cette méthode est bien celui du réordonnancement. Dans le cas évoqué ci-dessus, puisque la grammaire est probabilisée, on peut associer un poids $P(.)$ à chaque élément de l'ensemble $Gen(x)$. La méthode transforme alors $P(.)$ en $F(.)$. Les poids associés par $P(.)$ aux éléments de $Gen(x)$ permettent de les classer ; il en est de même pour $F(.)$. Cette méthode remplace donc le classement induit par $P(.)$ par le classement induit par $F(.)$. Ajoutons également que cette méthode, reposant sur le couplage de deux mécanismes, est fortement non inversible.

Apprentissage statistique (paramétrique) Une approche consiste à exprimer explicitement la dépendance entre l'entrée et la sortie sous la forme d'une distribution de probabilité $\mathbb{P}(x, y)$. Dans ce contexte, les deux problèmes les plus fréquemment étudiés sont l'apprentissage de données séquentielles et le parsage (*parsing* en anglais). Dans le premier, chaque élément (x, y) de la base d'apprentissage est un couple de séquences de même taille. Dans le second, x est une séquence et y un arbre d'analyse de cette séquence. La distribution $\mathbb{P}$ dépend généralement d'un ensemble de paramètres qu'il faudra estimer à partir d'un corpus de données, d'où le terme d'apprentissage statistique paramétrique.

Dans la suite, nous étudions le cas de l'apprentissage de données séquentielles. Dans ce problème, un scénario courant repose sur la métaphore dite du « canal bruité ». Dans celui-ci, on cherche à inférer une séquence de *variables (ou états) cachées* y, à l'aide d'une séquence de *variables observées* x. La séquence y constitue la « source », et x le « message reçu ». Par exemple, dans une application de

transcription orthographique/phonétique, y représente une séquence phonétique $y_1 \dots y_m$ et x une séquence graphique $x_1 \dots x_m$. On suppose alors que l'on dispose d'un modèle génératif de la source (permettant de définir $\mathbb{P}(y)$), et d'un modèle caractérisant la façon dont la source émet le message observé (i.e. $\mathbb{P}(x|y)$) ; il est alors possible d'exprimer $\mathbb{P}(x, y)$ selon $\mathbb{P}(x, y) = \mathbb{P}(x|y)\mathbb{P}(y)$. Pour déterminer la source à l'origine du message x, une stratégie consiste à maximiser relativement à y la quantité $\mathbb{P}(x, y) = \mathbb{P}(x|y)\mathbb{P}(y)$.

De manière à pouvoir manipuler ces grandeurs, des hypothèses doivent être effectuées. Par exemple, les modèles de Markov cachés (*Hidden Markov Models* (HMM) en anglais, voir Rabiner & Juang (1986)) imposent un certain nombre de contraintes sur la distribution $\mathbb{P}$. En particulier, elle doit vérifier :

- $\mathbb{P}(y_i|y_1 \dots y_{i-1}) = \mathbb{P}(y_i|y_{i-1})$ (propriété markovienne d'ordre 1) ;
- $\mathbb{P}(y_i|y_{i-1})$ ne dépend pas de i mais uniquement des valeurs y_i et y_{i-1} (stationnarité) ;
- $\mathbb{P}(x|y) = \prod_i \mathbb{P}(x_i|y_i)$ (indépendance des éléments de x conditionnellement à y).

Sous ces conditions, on peut écrire [5] :

$$\mathbb{P}(x|y)\mathbb{P}(y) = \prod_i \mathbb{P}(x_i|y_i)\mathbb{P}(y_i|y_{i-1}).$$

Un HMM est donc entièrement décrit par les paramètres $\mathbb{P}(x_i|y_i)$ (probabilité d'émettre x_i si l'état de la source est y_i), et les paramètres $\mathbb{P}(y_i|y_{i-1})$ (probabilité de transiter de l'état y_{i-1} à l'état y_i). De nombreuses techniques existent pour traiter le problème de l'évaluation (calculer $\mathbb{P}(x)$, la probabilité d'observer la séquence x), du décodage (trouver $argmax_y \mathbb{P}(y|x)$, la séquence d'états y la plus probable étant donnée la séquence observée x), et de l'estimation des paramètres (calculer les paramètres du HMM maximisant $\mathbb{P}(x, y)$) [6].

Les principales limitations des HMM résident dans les contraintes très fortes qu'ils imposent. En particulier, la propriété markovienne (d'ordre 1) empêche d'exprimer des dépendances entre états éloignés : la dépendance entre y_i et y_{i+3} passe nécessairement par y_{i+1} et y_{i+2}. En outre, la quantité $\mathbb{P}(y_i|y_{i-1})$ ne dépend pas de l'observation x_i, et l'émission de celle-ci est uniquement conditionnée par l'état courant y_i. Il est ainsi impossible de modéliser une dépendance entre l'observation x_i et des états plus lointains, dans le passé ($y_j, j < i$) comme dans l'avenir ($y_j, j > i$).

De tels modèles sont dits *génératifs* car ils supposent que (le message) x a été généré par (la source) y, génération explicitée par $\mathbb{P}(x|y)$. D'autres directions envisagent de maximiser directement (en y) la quantité $\mathbb{P}(y|x)$, sans passer par un modèle génératif de x. En d'autres termes, il n'est pas nécessaire de comprendre comment x a été produit : il suffit d'être en mesure de *discriminer* entre les y, c'est-à-dire savoir choisir la séquence y la plus adaptée à une séquence x donnée.

Les champs aléatoires conditionnels (*Conditional Random Fields* (CRF) en an-

5. Dans cette expression, la quantité $\mathbb{P}(y_1|y_0)$ exprime la probabilité que l'état y_1 soit *initial*.

6. Les algorithmes les plus connus pour traiter le deux premiers problèmes sont respectivement l'algorithme *forward* et l'algorithme de *Viterbi*.

glais), introduits par Lafferty *et al.* (2001), constituent un tel modèle exprimant directement $\mathbb{P}(y|x)$. Là encore, pour rendre possibles les calculs, un certain nombre d'hypothèses doivent être formulées. Dans le cas des CRF, il s'agit également d'une hypothèse de type markovien[7] :

$$\mathbb{P}(y_i|x, y_{-i}) = \mathbb{P}(y_i|x, y_{i-1}).$$

Sous cette hypothèse, il est possible de montrer (Lafferty *et al.*, 2001) que $\mathbb{P}(y|x)$ peut s'exprimer sous la forme :

$$\mathbb{P}(y|x) \propto \prod_i M_i(y_i, y_{i-1}, x),$$

avec

$$M_i(y_i, y_{i-1}, x) = exp(\sum_j \lambda_j t_j(y_i, y_{i-1}, x, i) + \sum_k \mu_k s_k(y_i, x, i)),$$

où λ_j et μ_k sont des paramètres réels et t_j et s_k sont des fonctions (souvent choisies booléennes) exprimant des propriétés sur les séquences. Par exemple, dans une tâche d'étiquetage morpho-syntaxique, une telle propriété peut être :

$$s_k(y_i, x, i) = \begin{cases} 1 & \text{si } y_i = \textit{Verbe}, x_{i-1} = \textit{ne}, \text{ et } x_{i+1} = \textit{pas}, \\ 0 & \text{sinon.} \end{cases}$$

En théorie, les CRF peuvent ainsi exploiter des propriétés globales ; la fonction $s_k(y_i, x, i)$ dépend de l'intégralité de la séquence x. Cette spécificité rend les CRF beaucoup plus souples que les HMM. En pratique, il faut un peu modérer ce discours car la difficulté des calculs conduit rapidement à simplifier les modèles. Sur les CRF, voir aussi Wallach (2004).

De manière générale, les problèmes d'apprentissage cherchent à modéliser une correspondance entre des entrées et des sorties possibles. Les approches statistiques paramétriques, qu'elles soient génératives ou discriminantes, abordent le problème de l'apprentissage de données séquentielles à l'aide d'une hypothèse commune : cette correspondance peut être décomposée en correspondances entre entités plus petites. Ainsi, pour résoudre une tâche de prononciation, les HMM utilisent une correspondance entre les graphèmes et les phonèmes, modélisée par les paramètres $\mathbb{P}(x_i|y_i)$. Cette décomposition est clairement visible dans les formules exprimant les HMM et les CRF, qui impliquent un produit indexé par la position dans les séquences.

Dans le cas d'un apprentissage de changement de niveau de représentation, cela signifie que l'on peut établir une correspondance *inter-niveaux* entre ces décompositions. Cette hypothèse est très forte ; nous la rediscuterons après avoir présenté un modèle permettant d'exploiter prioritairement les liens *intra-niveaux*.

cf. Sec. 3.4, p. 65

7. L'hypothèse posée est en réalité plus générale, car elle s'applique non seulement aux chaînes de Markov, mais à tout champ de Markov.

Résoudre le problème générique La voie la plus ambitieuse attaque directement le problème comme une tâche générique d'apprentissage automatique, en faisant le moins d'hypothèses possible sur la nature des objets. Cette problématique connaît un intérêt grandissant, comme en témoigne l'atelier récent *Learning with Structured Outputs* à la conférence NIPS (2004) et le programme thématique du réseau européen Pascal (2006), *Learning with Complex and Structured Outputs*. La bioinformatique et le TAL font partie des domaines d'application prioritairement visés par ce type d'approches. Cependant, bien que plusieurs modèles issus des *méthodes à noyaux* (Schölkopf & Smola, 2002), de plus en plus répandues dans la communauté de l'Apprentissage Automatique, aient été proposés de manière à pouvoir manipuler, dans l'espace d'entrée, des objets structurés, tels que les séquences de symboles, les arbres ou les graphes (cf. par exemple Haussler (1999) ; Collins & Duffy (2001) ; Kashima & Koyanagi (2002) ; Suzuki *et al.* (2003) ; Gärtner *et al.* (2004)), le problème des sorties structurées reste entier. Citons néanmoins Weston *et al.* (2003), Tsochantaridis *et al.* (2004) et Taskar *et al.* (2004), qui fournissent des éléments pour l'étude de ces problèmes non résolus [8].

3.2 Un apprenti paresseux à base de proportions analogiques

3.2.1 Introduction

Les réponses couramment proposées au problème de l'apprentissage automatique de changement de niveau de représentation ne reposent pas sur l'apprentissage par analogie, décrit et motivé dans le chapitre 2. Dans un tel type d'apprentissage, l'inférence s'effectue directement à l'aide d'exemples mémorisés, et aucune abstraction de données n'est formulée ; il repose également sur l'exploitation de proportions analogiques. Parmi les réponses relevées ci-dessus, la démarche qui se rapproche le plus d'un apprentissage par analogie est celle adoptée par l'équipe de W. Daelemans. Elle consiste à (i) transformer tout problème en un problème de classification, (ii) appliquer une méthode d'apprentissage paresseux sur ce nouveau problème de classification. Les apprentis paresseux reposent sur un apprentissage par similarité, version dégénérée de l'apprentissage par analogie.

cf. Sec. 2.2.3, p. 28

Cette méthode présente principalement deux limitations. Premièrement, nous avons souligné dans la section 3.1 la non adaptation des méthodes de classification aux problèmes d'apprentissage automatique de changement de niveau de représentation linguistique ; transformer un tel problème en une tâche de classification requiert une phase importante de pré-traitement des données. Ce pré-traitement n'est pas naturel lorsque les données sont complexes (arbres, structures de traits,

8. La solution proposée par Weston *et al.* (2003) se rapproche en réalité de celles de Collins & Koo (2005). En effet, même si le problème visé est générique, leurs méthode permet principalement d'affecter un poids à une solution possible. Le problème résolu est donc celui du classement de solutions. Selon les cas, il est possible ou non d'effectuer le calcul explicite de la solution pour laquelle ce poids est minimal.

etc.). Deuxièmement, la simple similarité ne permet pas de prendre véritablement en compte la nature structurelle des objets manipulés (relations, dépendances, etc.). Dans la suite, nous présentons une méthode d'apprentissage par analogie, reposant sur la notion de proportion, et répondant à ces deux limitations.

3.2.2 Des voisins aux proportions

Rappelons qu'une proportion analogique est une relation impliquant quatre objets x, y, z et t, notée $x : y :: z : t$ et se lisant « x est à y ce que z est à t ». Quand le dernier terme est manquant, nous parlons d'équation analogique. La résolution de l'équation $x : y :: z :?$ consiste à trouver tous les t tels que $x : y :: z : t$. L'ensemble des solutions de cette équation sera noté $S(x : y :: z :?)$. Nous fournissons dans le chapitre 4 des modèles formels de telles proportions analogiques, permettant d'appliquer dans de nombreux contextes la méthode présentée dans la suite, décrite originellement par Yvon (1996, 1997) et Pirrelli & Yvon (1999). Nous fournissons également des algorithmes permettant de vérifier si quatre termes forment une proportion, et de trouver les solutions d'une équation analogique. Dans la suite, nous faisons l'hypothèse qu'un mécanisme de construction et de résolution d'équation analogique est disponible [9].

Confronté à une nouvelle situation, l'apprentissage par analogie effectue une recherche de situations mémorisées analogues, et apparie la nouvelle situation avec une ou plusieurs situations passées, permettant un transfert de connaissances de ces dernières vers la première. Dans l'algorithme des k-ppv, si x est l'objet à analyser, la recherche d'« analogues » consiste à trouver l'ensemble $N_k(x)$ des k plus proches voisins de x dans une base d'apprentissage BA, où la notion de proximité est définie relativement à une certaine mesure de similarité. L'étape de transfert consiste à prédire $f(x)$ à partir de $f(N_k(x))$.

Dans la section 2.2.1, nous avions schématisé cette étape de transfert par la résolution d'équations analogiques telles que :

$$\begin{array}{rcl} x & : & y \\ & :: & \\ f(x) & : & ? \end{array} \quad \Rightarrow f(y) = f(x)$$

La même équation intervient dans un contexte de raisonnement à partir de cas. Dans ce cadre, on distingue généralement les approches transformationnelles des approches dérivationnelles (cf. figure 3.1).

cf. Sec. 2.1.1, p. 14

Dans les premières, on cherche à modéliser la « transformation » de x à y, et d'appliquer la même transformation à $f(x)$, de manière à obtenir $f(y)$. La seconde approche consiste à l'inverse à modéliser une dérivation de x à $f(x)$ et à l'appliquer

9. Le lecteur désirant connaître nos propositions concernant la définition et le calcul de ces proportions pourra lire préalablement le chapitre 4, relativement indépendant de celui-ci. Nous avons délibérément choisi de présenter dans un premier temps le modèle d'apprentissage, afin que la définition de la notion de proportion analogique devienne un besoin.

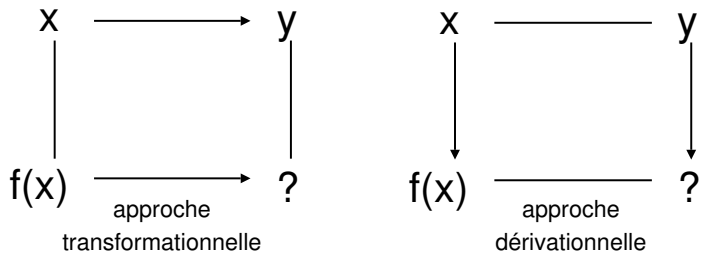


FIG. 3.1 – Approches transformationnelle et dérivationnelle

à y. Ces approches supposent de savoir lier de tels objets entre eux. En raisonnement à partir de cas, la nature des objets considérés permet d'envisager naturellement ces opérations. En particulier, les problèmes et les solutions peuvent être formulés à l'aide d'un même formalisme (réseaux sémantiques, trames, logiques de descriptions, etc.), rendant l'équation homogène. En outre, il est courant d'utiliser des connaissances spécifiques à la tâche à résoudre.

Dans le cas des k-ppv, l'équation est non-homogène ; les entrées et les sorties n'appartiennent pas au même espace. Puisque l'espace de sortie est relativement simple (un ensemble fini de classes), la question de la transformation ou de la dérivation ne se pose pas : la seule réponse envisageable est $f(x)$.

Pour considérer une équation homogène tout en ne faisant pas d'hypothèse sur (i) la nature des objets manipulés, (ii) la façon dont les entrées et les sorties sont reliées, on peut poser le schéma suivant (Yvon, 1996, 1997 ; Pirrelli & Yvon, 1999).

$$\begin{array}{ccc} (x, f(x)) & : & (y, f(y)) \\ & :: & \\ (z, f(z)) & : & (t, ?) \end{array}$$

Il s'agit ici d'exploiter la notion de proportion analogique dans l'espace d'entrée ($x : y :: z : t$), et dans l'espace de sortie ($f(x) : f(y) :: f(z) :?$) ; dans ce cadre, quels que soient les espaces d'entrée et de sortie, les proportions et les équations sont homogènes. En outre, l'introduction des proportions dans chacun des espaces permet de rendre compte des relations entretenues par les objets considérés. Nous dérivons de ces considérations la méthode présentée dans la suite.

Par ailleurs, de façon à mettre en évidence la propriété de symétrie des problèmes que l'on désire traiter, à savoir ceux impliquant un changement de niveau de représentation, un objet de l'espace d'entrée sera noté $i(x)$ et la sortie associée $o(x)$; x représente ici l'entité linguistique abstraite, et $i(x)$ et $o(x)$ correspondent à deux représentations de celle-ci. Le schéma proposé devient alors :

$$\begin{array}{ccc} (i(x), o(x)) & : & (i(y), o(y)) \\ & :: & \\ (i(z), o(z)) & : & (i(t), ?) \end{array}$$

3.2.3 APPA

Tout d'abord, les exemples de la base d'apprentissage sont stockés ; aucun traitement supplémentaire n'est effectué. C'est la caractéristique de l'apprentissage paresseux (Aha *et al.*, 1991 ; Aha, 1997).

Ensuite, étant donnée une nouvelle entrée $i(t)$ à analyser, la phase de recherche d'analogues consiste à identifier des proportions analogiques dans l'espace d'entrée impliquant $i(t)$ et d'autres objets contenus dans la base d'apprentissage, de façon à obtenir des triplets $(i(x_i), i(y_i), i(z_i))$ tels que $i(x_i) : i(y_i) :: i(z_i) : i(t)$. Cette recherche de proportions permet de capturer implicitement les liens structurels entre l'objet à analyser $i(x)$ et les objets connus.

Puis, pour chacun de ces triplets, la proportion $o(x_i) : o(y_i) :: o(z_i) : o(t)$ est supposée vérifiée. Toute solution de l'équation $o(x_i) : o(y_i) :: o(z_i) :?$ est alors un candidat potentiel pour $o(t)$. La réponse pour $o(t)$ est finalement effectuée en agrégeant ces différents candidats, par exemple en retenant le plus fréquemment proposé. Par conséquent, nous avons prédit $o(t)$ à partir des solutions des équations $o(x_i) : o(y_i) :: o(z_i) :?$. Cet algorithme, baptisé APPA (Apprentissage Paresseux par exploitation de Proportions Analogiques), est décrit de façon plus détaillée ci-dessous.

entrée : Un objet $i(t)$
sortie : Une proposition $o(t)$
begin
  $\mathcal{T} \longleftarrow \{(i(x), i(y), i(z)) \in BA^3_X \mid i(x) : i(y) :: i(z) : i(t)\}$
  $H \longleftarrow \emptyset$
  foreach $(i(x), i(y), i(z)) \in \mathcal{T}$ **do**
    $H \longleftarrow H \cup S(o(x) : o(y) :: o(z) :?)$
  end
end
return agrégation(H)

Algorithme 1 : APPA

Hypothèse analogique L'« hypothèse analogique » sous-jacente à la méthode s'exprime de la façon suivante (cf. figure 3.2).

Si « $i(x)$ est à $i(y)$ ce que $i(z)$ est à $i(t)$ »,
alors « $o(x)$ est à $o(y)$ ce que $o(z)$ est à $o(t)$ ».

Cette hypothèse exprime le fait qu'une proportion analogique dans l'espace d'entrée est conservée dans l'espace de sortie et repose sur l'idée générale que les relations entretenues entre les représentations dans l'espace d'entrée le restent dans l'espace de sortie. Nous reviendrons sur la nature de cette hypothèse un peu plus loin. Illustrons pour l'instant cet algorithme à l'aide du cas suivant.

cf. Sec. 3.3, p. 58

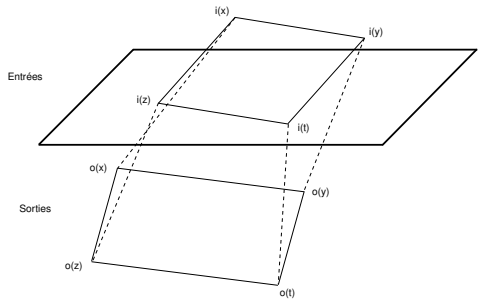


FIG. 3.2 – Hypothèse analogique

Déroulement de l'algorithme sur un exemple Nous disposons d'une base d'apprentissage [10] :

$$BA = \{e_1 = (parler, 2, parles), e_2 = (parler, PP, parlant), e_3 = (regarder, 2, regardes),$$
$$e_4 = (regarder, PP, regardant), e_5 = (sembler, 2, sembles)\},$$

et nous devons analyser l'exemple $e_6 = (sembler, PP, ?)$, dont le troisième attribut est non connu (symbole '?'). L'espace d'entrée correspond aux deux premiers attributs, et l'espace de sortie au troisième. La première étape consiste à rechercher les triplets formant avec e_6 une proportion analogique dans l'espace d'entrée. On peut aisément vérifier que cette recherche fournit pour résultat les deux triplets $(i(e_1), i(e_2), i(e_5))$ et $(i(e_3), i(e_4), i(e_5))$. Ensuite, il s'agit de résoudre les équations analogiques suivantes :

$$o(e_1) : o(e_2) :: o(e_5) :? \text{ et } o(e_3) : o(e_4) :: o(e_5) :?, \text{ c'est-à-dire}$$

$$parles : parlant :: sembles :? \text{ et } regardes : regardant :: sembles :?.$$

La résolution de ces équations conduit à la solution *semblant*, alors proposée pour l'entrée $(sembler, PP, ?)$ [11]. Dans le cas général, plusieurs solutions peuvent être trouvées et des stratégies d'agrégation doivent être définies.

cf. Sec. 5.1.2, p. 123

Niveaux de représentation et symétrie entrée/sortie Nous avons jusqu'ici étudié le passage d'un niveau de représentation à un autre : le premier correspond à l'espace d'entrée et le second à l'espace de sortie. En réalité, à partir d'un même jeu de données, plusieurs tâches différentes sont envisageables. En particulier, il est possible de considérer le problème général d'apprentissage supervisé suivant.

Soit une base d'apprentissage $BA = \{e_1, \ldots, e_n\}$ dans laquelle chaque exemple e_i est un vecteur constitué de m attributs $\langle e_{i1}, \ldots, e_{im} \rangle$. Ici, chaque attribut est de

10. Dans cette base, 2 représente schématiquement la deuxième personne du présent de l'indicatif et *PP* le participe présent.

11. Cette affirmation sera justifiée dans le chapitre suivant qui traite notamment de la résolution formelle de ce type d'équation sur les chaînes.

nature quelconque (séquence, arbres, vecteurs, etc.). Étant donnée la base d'apprentissage BA et un nouvel exemple $e \notin BA$ dont les attributs ne sont pas tous informés, la tâche d'apprentissage consiste à prédire les attributs non connus de e. L'ensemble des attributs connus (resp. inconnus) de e forme donc l'*espace d'entrée* X (resp. l'*espace de sortie* Y). La partie connue (resp. inconnue) de e sera notée $i(e) \in X$ (resp. $o(e) \in Y$) dans la suite. Chaque attribut $k \in \{1, \ldots, m\}$ peut correspondre à un niveau de représentation. En prenant $m = 2$, nous pouvons modéliser le passage d'un niveau de représentation à un autre. Si davantage d'information est disponible, on peut choisir $m > 2$ pour prendre en compte simultanément plusieurs niveaux de représentation et croiser les connaissances. De plus, les espaces d'entrée et de sortie ne sont pas posés a priori, mais déterminés par les attributs manquants de l'exemple à analyser. Cette propriété nous permet d'envisager naturellement des problèmes inverses l'un de l'autre (entrée $\leftrightarrow$ sortie). Par exemple, pour la base d'apprentissage

$$BA = \{e_1 = (parler, 2, parles), e_2 = (parler, PP, parlant), e_3 = (regarder, 2, regardes),$$
$$e_4 = (regarder, PP, regardant), e_5 = (sembler, 2, sembles)\},$$

si l'objet à analyser est $(?, ?, semblant)$, la recherche de triplets dans l'espace d'entrée fournit $(i(e_1), i(e_2), i(e_5))$ et $(i(e_3), i(e_4), i(e_5))$, avec maintenant $i(e_1) = parles$, $i(e2) = parlant$, etc. Ensuite, il s'agit de résoudre les deux équations analogiques

$$o(e_1) : o(e_2) :: o(e_5) :? \text{ et } o(e_3) : o(e_4) :: o(e_5) :?, \text{ soit}$$
$$(parler, 2) : (parler, PP) :: (sembler, 2) :? \text{ et}$$
$$(regarder, 2) : (regarder, PP) :: (sembler, 2) :?,$$

dont la solution est $(sembler, PP)$ [12].

La même base d'apprentissage peut donc être utilisée pour des tâches d'apprentissage différentes, ce qui rend la méthode relativement souple. Par exemple, on peut envisager de traiter simultanément les problèmes de l'analyse et de la génération de données morphologiques.

Indépendance entrée/sortie En outre, la dépendance entre l'entrée et la sortie n'est jamais établie de façon directe, mais passe toujours par l'intermédiaire des objets impliqués dans les proportions. Par exemple, le fait qu'une partie de l'entrée *(sembler, PP)* et la sortie proposée *semblant* partagent une sous-chaîne de symboles n'intervient jamais dans le mécanisme d'apprentissage. Ainsi, pour la base d'apprentissage

$$BA = \{e_1 = (parler, 2, febo), e_2 = (parler, PP, febu), e_3 = (regarder, 2, pabo),$$
$$e_4 = (regarder, PP, pabu), e_5 = (sembler, 2, tebo)\},$$

la sortie proposée pour $(sembler, PP, ?)$ est *tebu*. Aucun alignement particulier n'est à effectuer entre les entrées et les sorties associées. Cela permet de traiter des objets de nature différente, et cela sans adaptation spécifique. Nous reviendrons sur cette propriété.

cf. Sec. 3.4, p. 65

12. Idem.

Objets structurés et « sortie » de BA_Y L'étape de transfert invoquée par APPA permet de « sortir » de BA_Y. En effet, la solution d'une équation $o(x) : o(y) :: o(z) :?$ peut très bien ne pas apparaître dans la base d'apprentissage ; c'est le cas de *semblant* dans l'exemple qui nous a servi d'illustration à la section 3.2.3. En outre, nous n'avons pas fait d'hypothèse sur la nature des données traitées ; le seul pré-requis est d'être en mesure de définir la notion de proportion analogique dans les espaces d'entrée et de sortie de manière à (i) pouvoir trouver les triplets impliqués dans une proportion dans l'espace d'entrée, (ii) savoir résoudre les équations analogiques dans l'espace de sortie. Il est ainsi possible de traiter des objets structurés tels que les chaînes ou les arbres dès lors que l'on sait définir la notion de proportion entre de tels objets. La manipulation d'objets d'un degré de complexité quelconque peut ainsi être envisagée, bien que les procédures de recherche de triplets et de résolution d'équations impliquées risquent de présenter une complexité en conséquence. L'élément à retenir ici est que la même procédure générale s'utilise quelle que soit la nature des objets considérés.

Ambiguïtés et solutions multiples Soulignons enfin le fait que la procédure présentée permet de fournir plusieurs réponses à une entrée donnée, vérifiant ainsi le pré-requis concernant la gestion des ambiguïtés. En effet, plusieurs triplets peuvent être identifiés dans l'espace d'entrée, et chaque triplet conduit à une équation ayant potentiellement plusieurs solutions. L'algorithme des k-ppv présente également cette propriété. En revanche, alors que ce dernier propose une réponse quelle que soit l'entrée (de même que la plupart des apprentis rencontrés en apprentissage automatique supervisé), le modèle étudié peut ne fournir aucune réponse si aucun triplet n'est trouvé ou si les équations impliquées ne présentent pas de solutions.

Cela peut au premier abord sembler être une faiblesse du modèle. En réalité, ce comportement est parfaitement compatible avec de nombreuses tâches de traitement automatique des langues. En effet, d'une part il est tout à fait souhaitable de ne pas proposer d'analyse morpho-syntaxique pour une entrée telle que *jssqnlkkkh* [13], d'autre part on peut en attendre plusieurs pour *marche*, qui peut être un verbe ou un nom.

Principales limitations La procédure d'apprentissage par analogie développée présente donc les propriétés requises pour le traitement des tâches visées, à savoir : (i) elle peut manipuler des objets structurés, (ii) elle respecte la symétrie entrée/sortie des tâches visées, (iii) elle peut fournir des solutions non initialement contenues dans la base d'apprentissage. Elle soulève tout de même deux problèmes particuliers :

- Plusieurs hypothèses peuvent être proposées pour $o(t)$: un classement ou une sélection d'hypothèses doit être effectué ;
- L'exploration exhaustive de BA_X^3 conduit à l'évaluation de $|BA_X|^3$ triplets, limitant le traitement de larges bases de données.

13. Précisons : en français.

L'algorithme des *k*-ppv, de même que la plupart des méthodes d'apprentissage paresseux, est confronté également à ces deux problèmes. En ce qui concerne le premier, une simple procédure de vote, éventuellement pondéré, peut s'appliquer. Nous présentons plus loin un certain nombre de stratégies d'agrégation. Le second point est en revanche plus problématique. Pour les apprentis paresseux faisant intervenir une distance, de nombreuses méthodes ont été développées de manière à optimiser cette recherche, (i) en réduisant la taille de la base d'apprentissage, (ii) en organisant efficacement les exemples de la base d'apprentissage, (iii) en exploitant certaines propriétés au moment de la recherche. Les techniques dites de réduction ou de condensation (Hart, 1968 ; Gates, 1972 ; Aha *et al.*, 1991) ont pour but de supprimer des exemples de la base d'apprentissage, tout en ne nuisant pas aux performances. On pourra, par exemple, supprimer des instances « inutiles », comme celles largement entourées de voisins de la même classe. Les techniques d'édition (Devijver & Kittler, 1980) ont pour objectif l'amélioration de la précision de l'algorithme, en supprimant les exemples considérés bruités, comme ceux largement entourés de voisins de classes différentes ; les frontières de séparation entre les classes sont de cette façon lissées. Dans ce cas, la diminution de la taille de la base d'apprentissage est un effet secondaire agréable. Sur ces techniques, voir Dasarathy (1990) ; Dasarathy *et al.* (2000) ; Wilson & Martinez (2000). Le volume occupé par les exemples peut également être réduit par une représentation efficace de ceux-ci, par exemple sous la forme d'arbres de recherche (Daelemans *et al.*, 1997*b*). Enfin, des méthodes telles qu'AESA (Micó *et al.*, 1994), exploitent un principe permettant de s'affranchir de l'étude de certains exemples dans la recherche des voisins.

cf. Sec. 5.1.2, p. 123

Dans la suite, nous introduisons la notion d'*extension analogique*, qui nous permet de modéliser les mécanismes d'apprentissage sous-jacents à APPA. Nous proposerons des réponses aux limitations évoquées ci-dessus ultérieurement.

cf. Sec. 5.1, p. 120

3.3 Extension analogique et biais d'apprentissage

3.3.1 Introduction

L'algorithme APPA est en mesure de proposer des solutions non initialement contenues dans la base d'apprentissage ; la fonction de décision g lui étant associée vérifie donc $g(X) \supseteq BA_Y$. Dans cette section, nous caractérisons plus précisément ce comportement, à l'aide de l'introduction de la notion d'extension analogique. Nous montrons ensuite comment cette extension correspond à un biais d'apprentissage permettant à APPA d'effectuer une inférence.

3.3.2 Extension analogique

L'extension analogique permet de modéliser la *création analogique*, c'est-à-dire la construction de nouveaux objets par résolution d'équation analogique.

Définition 1 (Extension analogique). L'*extension analogique* $E_A(E)$ d'un ensemble E contient les éléments formant une proportion analogique avec trois éléments de E :

$$E_A(E) = \{t \mid x : y :: z : t \text{ avec } (x, y, z) \in E^3\},$$

ce qui donne, en notant $S(x : y :: z :?)$ l'ensemble des solutions de l'équation analogique $x : y :: z :?$,

$$E_A(E) = \cup_{(x,y,z)\in E^3} S(x : y :: z :?).$$

En itérant, on obtient une extension d'*ordre* quelconque.

Définition 2 (Extension analogique (d'ordre n)). L'*extension analogique d'ordre n* (entier naturel) est définie par récurrence :

- $E_A^0(E) = E$;
- $E_A^n(E) = E_A(E_A^{n-1}(E))$ si $n > 0$.

Les extensions forment une suite d'ensemble emboîtés. En effet, puisqu'on a

$$\forall a, a : a :: a : a,$$

on a également

$$E \subseteq E_A(E),$$

et plus généralement,

$$\forall n \in \mathbb{N}, E_A^n(E) \subseteq E_A^{n+1}(E).$$

Nous pouvons donc poser :

$$E_A^\infty(E) = \cup_{n\in\mathbb{N}} E_A^n(E).$$

L'extension analogique [14] (EA) d'un ensemble E correspond donc à l'ensemble des objets pouvant être construits par analogie à partir des éléments de E. Les éléments de cette extension analogique sont, comme le fait remarquer justement Saussure pour le langage, déjà présents en puissance dans E.

> Un mot que j'improvise, comme *in-décor-able*, existe déjà en puissance dans la langue ; on retrouve tous ses éléments dans les syntagmes tels que *décor-er : décor-ation*, *pardonn-able : mani-able*, *in-connu : in-sensé*, etc., et sa réalisation dans la parole est un fait insignifiant en comparaison de la possibilité de le former.
>
> de Saussure (1916, p. 227)

Ce comportement est illustré sur les figures 3.3 et 3.4.

Il est également possible de définir une notion duale de l'extension analogique, notion que l'on appellera *support analogique*. Un support analogique est défini comme étant un sous-ensemble capable de reconstruire un ensemble par analogie. Un tel support n'est pas nécessairement unique.

14. Voir également Lepage (2000) pour une notion apparentée.

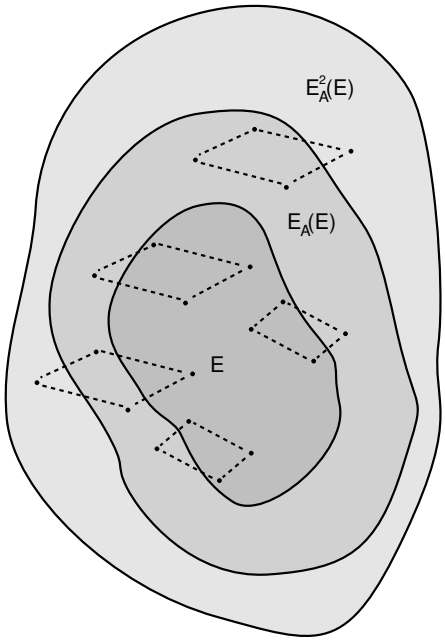


FIG. 3.3 – Extension analogique

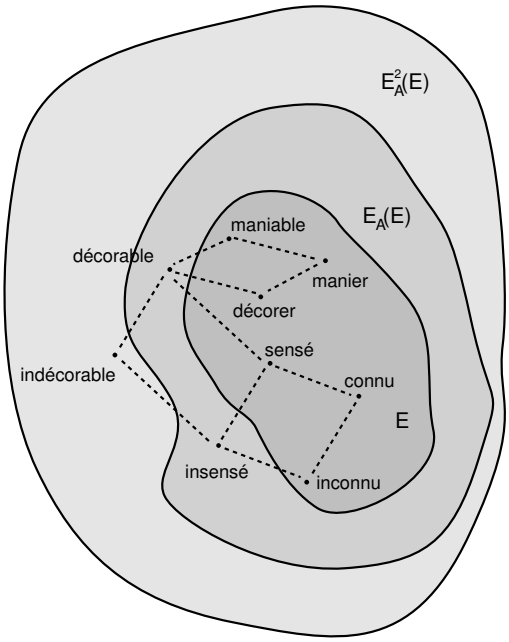


FIG. 3.4 – Extension analogique : cas de *indécorable*

Définition 3 (Support analogique). A_S est un *support analogique* de E si et seulement si

$$E_A(A_S) \supseteq E.$$

Il est dit minimal si

$$\forall A \subseteq A_S, E_A(A) \not\supseteq E.$$

Le support analogique d'un ensemble correspond donc à une sorte de cœur structurel. De la même façon que l'extension analogique d'un ensemble est en puissance dans celui-ci, un ensemble est en puissance dans son support analogique.

Apprentissage par complétion Les éléments de l'extension analogique de E, déjà en puissance dans E, sont donc soumis à ce que nous avions rapidement évoqué dans la section 2.1.1 sous l'expression « pression d'existence ». Nous avions alors expliqué, de façon intuitive, qu'un appariement entre deux objets a et b d'une part et une relation entre a et c d'autre part, suggéraient l'existence d'un élément d tel que $a : b :: c : d$. L'extension analogique permet donc de modéliser la phase de transfert impliquée dans un raisonnement par analogie. Cette phase permet de compléter une connaissance sur un domaine cible, par une adaptation des éléments inappariés du domaine source vers le domaine cible. Cette complétion peut être modélisée à l'aide de la notion d'extension analogique : en notant D_1 et D_2 les ensembles d'éléments des domaines source et cible, la connaissance acquise sur D_2 s'obtient par l'extension analogique $E_A(D_1 \cup D_2)$. Dans l'exemple étudié dans la section 2.1, puisque l'on a *supérieur_1* : *cause_1* :: *supérieur_2* : *cause_2*, et *supérieur_1* : *flux_1* :: *supérieur_2* : *flux_2*, les éléments inférés *cause_2* et *flux_2*, appartiennent bien à l'extension analogique des deux domaines. Cette modélisation permet en outre de rétablir une certaine symétrie entre les domaines source et cible. En effet, le domaine source a été défini comme étant celui pour lequel le plus d'information est disponible. Dans un cas général, l'apprentissage est le résultat d'une interaction et d'un enrichissement mutuel ; le processus est alors moins clairement directionnel.

Les mécanismes sous-jacents à APPA exploitent l'extension analogique de manière à effectuer un « apprentissage par complétion ». L'hypothèse effectuée en réalité par cet algorithme est que l'exemple t à analyser est contenu dans l'extension analogique de la base d'apprentissage. Si tel est le cas, alors il existe un triplet (x, y, z) tel que $x : y :: z : t$. Or, on a [15] :

$$x : y :: z : t \Rightarrow i(x) : i(y) :: i(z) : i(t),$$

et

$$x : y :: z : t \Rightarrow o(x) : o(y) :: o(z) : o(t).$$

L'algorithme fonctionne ensuite par abduction, puisqu'il repose sur

$$i(x) : i(y) :: i(z) : i(t) \Rightarrow o(x) : o(y) :: o(z) : o(t),$$

ou plus exactement sur

$$i(x) : i(y) :: i(z) : i(t) \Rightarrow x : y :: z : t \Rightarrow o(x) : o(y) :: o(z) : o(t).$$

La proportion $i(x) : i(y) :: i(z) : i(t)$ est donc utilisée comme indice d'une proportion sur les objets réels. Si cette proportion existe sur l'intégralité des objets réels,

15. Ces implications résultent du fait que si quatre objets forment une proportion, alors c'est également le cas pour des projections de ceux-ci. Elles deviendront claires à la lecture du chapitre 4.

alors elle est trouvée par APPA (cf. tableau 3.4). Si elle existe uniquement dans l'espace d'entrée, alors l'approximation effectuée par l'abduction conduit potentiellement à la proposition de mauvaises solutions. Si la proportion n'existe pas dans l'espace d'entrée, alors aucune solution n'est proposée ; l'algorithme est par conséquent assez conservateur et propose des solutions uniquement lorsqu'une forte structure relationnelle est identifiée (proportions à la fois dans les espaces d'entrée et de sortie).

	$o(t) \in \mathcal{O}(i(t))$	$o(t) \notin \mathcal{O}(i(t))$	
		$\mathcal{O}(i(t)) = \emptyset$	$\mathcal{O}(i(t)) \neq \emptyset$
$i(t) \in E_A(BA_X)$	bonne solution	pas de solution	risque de mauvaises solutions
$i(t) \notin E_A(BA_X)$	pas de solution		

$\mathcal{T}(i(t)) = \{(x_i, y_i, z_i) \in BA |\ i(x_i) : i(y_i) :: i(z_i) : i(t)\}$ et
$\mathcal{O}(i(t)) = \cup_{(x_i,y_i,z_i)\in\mathcal{T}(i(t))} S(o(x_i) : o(y_i) :: o(z_i) :?)$
(risque de mauvaises solutions : pas de solution ou mauvaise(s) solution(s))

TAB. 3.4 – Comportement d'APPA en terme d'extension analogique

Cette analyse rapide permet de mettre en évidence le fait que la notion d'extension analogique est au cœur du processus inférentiel invoqué par la méthode. Nous explorons cette idée de façon plus détaillée dans la suite.

3.3.3 Biais d'apprentissage

Inférence grammaticale D'un point de vue formel, toute fonction $f(.)$ peut être vue comme une fonction booléenne $f_b(.,.)$, avec $f_b(x, y) = \mathbb{1}_{[f(x)=y]}$. De même, tout problème d'apprentissage (quelle que soit la nature des objets impliqués en entrée et en sortie) associé à une distribution $\mathbb{P}(X, Y)$ et à la recherche d'une fonction $g(.)$ minimisant $L(g) = \mathbb{P}\{g(X) \neq Y\}$ peut être vu comme un problème de classification binaire où la fonction cible est g_b minimisant [16]

$$\mathbb{E}\{\mathbb{1}_{[g(X)\neq Y]}\} = \mathbb{E}\{1 - \mathbb{1}_{[g(X)=Y]}\} = \mathbb{E}\{1 - g_b(X, Y)\} = L(g_b).$$

Pour une entrée x, les sorties proposées par un apprenti $h_b(.,.)$ sont les objets y tels que $h_b(x, y) = 1$. Dans ce problème de classification binaire, la base d'apprentissage BA contient des exemples (x_i, y_i) positifs, c'est-à-dire tels que $g_b(x_i, y_i) = 1$.

Toujours d'un point de vue formel, tout objet [17] est représentable par une séquence de symboles sur un certain alphabet Σ. L'ensemble de ces séquences est le monoïde libre $\Sigma^\star$. Diviser l'ensemble $\Sigma^\star$ en deux classes, contenant respectivement les éléments dits positifs et les éléments dits négatifs revient à caractériser

16. Il s'agit ici de considérations formelles, n'entrant pas en contradiction avec la remarque soulignant la difficulté pratique et/ou le manque de justification à reformuler un problème d'apprentissage quelconque en une tâche de classification.

17. Il faut lire ici : « tout objet manipulé par une machine de Turing ».

un sous-ensemble $L \subseteq \Sigma^\star$. Un tel sous-ensemble s'appelle un *langage*. L'*inférence grammaticale* est le domaine de l'apprentissage automatique dont l'objectif consiste à inférer des langages à partir d'exemples. Certaines des questions traitées dans un tel domaine sont [18] :

- est-il possible d'apprendre le langage L à partir d'exemples ?
- uniquement à partir d'exemples positifs ?
- est-ce que tous les langages d'une famille de langages donnée sont apprenables, etc. ?

Nous disposons aujourd'hui d'un certain nombre de résultats énonçant l'apprenabilité ou la non apprenabilité de certaines classes de langages, dans différents cadres d'inférence.

Prenons un exemple pour illustrer ce problème formel. Supposons que les exemples positifs du langage visé dont on dispose soient les séquences suivantes :

$$ab, aabb, aaabbb, aaaabbbb, aaaaabbbbb.$$

Un langage cohérent avec ces exemples est $\{(a^n b^n)_{n\in\mathbb{N}_+}\}$. Cependant, ce n'est pas l'unique langage cohérent avec ceux-ci ; le langage $\{(a^n b^n)_{n\in\mathbb{N}_+}\} \cup \{bba\}$ ainsi que $\{(a^n b^n)_{n\in\{1,\ldots,31\}}\}$ conviennent également. On « sent » bien que le fait de disposer uniquement d'exemples positifs constitue une grande limitation : des informations supplémentaires sont nécessaires pour qu'une décision soit prise sur le langage cible. Par exemple, le fait de disposer d'un nombre suffisant d'exemples à la fois positifs et négatifs permet de guider plus précisément la recherche. Si le langage visé est trop complexe, cette donnée peut ne pas être encore suffisante : on peut alors être amené à ne considérer que des langages d'un certain type, ce qui exclut a priori une partie de l'espace de recherche. Nous explicitons ci-dessous ce besoin général qu'ont les méthodes d'apprentissage de disposer d'une connaissance additionnelle aux exemples.

Biais d'apprentissage [19] Une méthode d'apprentissage automatique supervisé a pour objectif de construire un apprenti à partir d'une base d'apprentissage. De manière générale, puisqu'une telle base peut être cohérente avec un nombre illimité d'apprentis, la méthode d'apprentissage qui choisit un apprenti doit reposer sur un critère exogène aux données [20]. Dans le domaine de l'apprentissage automatique, ce critère est usuellement appelé *biais d'apprentissage*. La définition d'un tel biais est une condition nécessaire au bon fonctionnement de la méthode (Mitchell, 1980 ; Wolpert & Macread, 1997) ; sans lui, elle avance aveuglément dans l'espace des apprentis possibles. Ce biais formule une restriction ou une préférence sur l'ensemble des apprentis possibles, guidant ainsi la recherche. En outre, des hy-

18. Pour répondre à ces questions, il est bien entendu nécessaire de donner un sens à la notion d'apprenabilité ; différents cadres d'apprentissage coexistent, que nous ne détaillons pas ici. Voir par exemple Cornuéjols & Miclet (2002, chap. 7) et Dupont & Miclet (1998).

19. Dans la littérature, il est également question de *principe inductif* ; cette expression a le mérite d'éviter toute confusion avec le biais du compromis biais/variance. Nous n'évoquerons pas ce compromis, et il n'y aura aucune confusion possible pour nous.

20. « Data will never replace knowledge. » Bousquet (2003).

pothèses doivent permettre de lier les observations de la base d'apprentissage aux observations susceptibles d'être rencontrées à l'avenir.

> If there is no assumption on how the past is related to the future, prediction is impossible.
> If there is no restriction on the possible phenomena, generalization is impossible.
>
> Bousquet (2003)

En général, on considère que toutes les observations (passées et futures) sont issues d'une même distribution, et qu'elles sont générées de manière indépendante (indépendance des exemples et stationnarité du phénomène). Pour établir une préférence entre les apprentis, si l'on exclut les biais spécifiques à une connaissance sur un domaine particulier, on invoque habituellement un critère de « simplicité » (rasoir d'Occam). Celui-ci peut prendre différentes formes : norme dans un espace fonctionnel (régularisation), dimension ou capacité d'un modèle (minimisation du risque structurel), simplicité algorithmique (cadre PACS), etc.

Dans le cas des séparateurs linéaires (cf. section 2.2.1), le biais d'apprentissage s'exprime clairement : seuls les apprentis effectuant une séparation linéaire de l'espace sont considérés (restriction de l'espace des apprentis). Dans le cas des k-ppv, il n'est pas explicite, mais repose sur l'assertion générale suivante : des entrées similaires conduisent à des sorties similaires, i.e. si a est proche de b, alors $f(a)$ est un bon candidat pour $f(b)$. APPA est construit à partir d'une telle assertion informelle : les proportions analogiques dans l'espace d'entrée sont conservées dans l'espace de sortie, i.e. si « $i(x)$ est à $i(y)$ ce que $i(z)$ est à $i(t)$ », alors « $o(x)$ est à $o(y)$ ce que $o(z)$ est à $o(t)$ ». L'inférence grammaticale fournit un cadre permettant d'expliciter simplement et formellement le biais de cette méthode.

Apprentissage par complétion et lien avec l'inférence grammaticale Dans un contexte d'inférence grammaticale, la généralisation effectuée par APPA consiste à construire d'autres exemples positifs par extension analogique. L'hypothèse postulée est que si un objet (i_t, o_t) appartient à $E_A(BA)$, il est positif ; ainsi, o_t est une sortie acceptable pour l'entrée i_t. En d'autres termes, le principe général suivant est posé :

$$S \subseteq L \Rightarrow E_A(S) \subseteq L.$$

Si aucun autre critère n'est utilisé que celui-ci, alors le langage inféré à partir d'une base BA d'exemples positifs est $E_A(BA)$. Notons que puisque l'apprentissage effectué par APPA est paresseux, ce langage n'est jamais explicitement construit : l'algorithme permet uniquement de déterminer si oui ou non un exemple appartient au langage. Il est également théoriquement possible d'itérer ce processus, de manière à se rapprocher de $E_A^\infty(BA)$. Dans ce cas, on part d'une base d'exemples positifs, que l'on complète étape par étape ; c'est donc un apprentissage « par complétion », et il est effectué uniquement à partir d'exemples positifs. Le fait de procéder par étape est bien la marque de l'approche analogique, qui infère du particulier au particulier ; ainsi, même si l'ensemble $E_A^\infty(BA)$ peut correspondre à une forme de connaissance générale, celle-ci n'est jamais explicitement atteinte. Remarquons

en outre que le modèle ne dit rien des éléments n'appartenant pas à $E_A^\infty(BA)$; on peut considérer qu'ils sont négatifs par défaut, mais le modèle, lui, ne se prononce pas.

Ce détour par l'inférence grammaticale nous permet de formaliser l'idée selon laquelle le biais d'apprentissage sous-jacent à l'algorithme APPA repose essentiellement sur la notion d'extension analogique, et que c'est elle qui dirige l'inférence. Ce biais est très fort, car il permet d'effectuer une inférence à partir d'exemples positifs seuls, tâche dont nous avons souligné précédemment la difficulté dans un cadre non restreint.

3.4 Proportions analogiques et paradigmes

3.4.1 Introduction

Nous avons étudié précédemment (cf. section 3.2) un algorithme d'apprentissage par analogie, reposant sur la notion de proportion et offrant une réponse au problème de l'apprentissage du changement de niveau de représentation. Nous avons également mis en évidence son biais d'apprentissage à l'aide de la notion d'extension analogique (cf. section 3.3). Nous revenons maintenant sur les fondements d'un tel biais, en montrant comment les proportions peuvent exploiter l'organisation paradigmatique des données linguistiques.

3.4.2 Paradigmes morphologiques

Morphologie flexionnelle Dans cette section, le vocabulaire utilisé est celui de la morphologie lexématique (Matthews, 1974 ; Fradin, 2003). Dans ce contexte, l'unité de base est le *lexème*.

> Le lexème est une unité abstraite à quoi se rapporte un mot-forme ; cette abstraction concerne les variations de formes.
>
> Fradin (2003, p. 102)

Les variations flexionnelles sont des variations de formes ; elles permettent de véhiculer un certain nombre de descripteurs linguistiques (dits *traits flexionnels*) tels que le genre (masculin/féminin), le nombre (singulier/pluriel), la personne ($1^{re}/2^e/3^e$), le cas (génitif/datif/accusatif), et le temps (présent/passé/futur), etc. Les variations flexionnelles permettent d'instancier un lexème abstrait pour qu'il se *réalise* dans une construction (syntaxique). Ces formes sont les *formes fléchies* du lexème, constituant son *paradigme flexionnel*.

Un lexème est souvent désigné conventionnellement par l'une de ces formes, par exemple la forme masculin singulier d'un nom ou la forme infinitive d'un verbe. Par exemple, *jolies* est une forme du « lexème » `JOLI`, et *parlerons* une forme du « lexème » `PARLER`. La première véhicule les descripteurs `féminin+pluriel`, et la deuxième les descripteurs `pluriel+1+futur`.

Paradigmes flexionnels et proportions En français, la flexion intervient principalement pour les verbes, noms et adjectifs. Dans ce cas, le processus flexionnel est quasiment systématique : toute combinaison valide de descripteurs conduit à une forme[21] [22] ; une telle combinaison correspond à une *case* du paradigme. Une combinaison d'un sous-ensemble de descripteurs conduit à un *paradigme partiel* ; le paradigme partiel du verbe *aimer* pour la combinaison `présent+indicatif` est illustré sur le tableau 3.5.

`présent+indicatif`					
`singulier`			`pluriel`		
1	2	3	1	2	3
j'aime	tu aimes	il aime	nous aimons	vous aimez	ils aiment

TAB. 3.5 – Un paradigme partiel du verbe `AIMER`

De tels paradigmes flexionnels sont très largement redondants ; autrement dit, il n'est pas nécessaire d'avoir accès à l'intégralité des paradigmes de l'ensemble des lexèmes existants pour pouvoir associer une forme et un ensemble de descripteurs. L'hypothèse analogique (des proportions dans un espace sont le signe de proportions dans un autre espace) fournit une modélisation de cette redondance : pour qu'un ensemble de paradigmes soit reconstructible par analogie, il « suffit » de disposer d'un support analogique de celui-ci. Cette approche est qualifiée de *paradigmatique* car elle s'appuie sur la structure des paradigmes ; en particulier, elle repose sur des proportions du type :

cf. Sec. 3.2.3, p. 54

cf. Sec. 3.3.2, p. 58

$$\begin{array}{lcl} \text{la case } i \text{ du paradigme de } P_1 & : & \text{la case } j \text{ du paradigme de } P_1 \\ & :: & \\ \text{la case } i \text{ du paradigme de } P_2 & : & \text{la case } j \text{ du paradigme de } P_2 \end{array} .$$

Par construction, une telle proportion s'observe dans l'espace des descripteurs, par exemple

$$\begin{array}{lcl} \texttt{PARLER;infinitif} & : & \texttt{PARLER;singulier+1+imparfait} \\ & :: & \\ \texttt{MARCHER;infinitif} & : & \texttt{MARCHER;singulier+1+imparfait} \end{array} .$$

Pour pouvoir véhiculer des descripteurs à partir de formes en s'appuyant sur l'hypothèse analogique (c'est-à-dire en utilisant l'algorithme APPA, cf. section 3.2), il faut que la proportion dans l'espace des formes soit identifiable à partir des structures seules de celles-ci ; c'est le cas de :

$$\begin{array}{lcl} \textit{parler} & : & \textit{je parlais} \\ & :: & \\ \textit{marcher} & : & \textit{je marchais} \end{array} .$$

21. Penser néanmoins au verbe *gésir* et aux noms n'existant qu'au pluriel comme *ténèbres*.

22. Les descripteurs dépendent de la catégorie grammaticale et de la langue. Par exemple, le descripteur *temps* n'est pas pertinent pour un nom. De même le descripteur *cas* ne conduit pas à des formes fléchies différentes en français, et la flexion des adjectifs anglais est très pauvre. La flexion nominale est souvent nommée déclinaison tandis que celle du verbe est appelée conjugaison.

Nous dirons qu'une telle proportion est *formelle* ou *structurelle* car elle ne nécessite aucun autre élément que la forme et la structure des termes qui la composent. La caractérisation de telles proportions formelles fera l'objet du chapitre 4.

Unités minimales et correspondance inter-niveaux Cette conception permet de faire l'économie (i) de la décomposition des représentations en unités minimales, (ii) de la correspondance entre les unités minimales d'un niveau et celles d'un autre niveau (correspondance *inter-niveaux*). Dans le cas de la flexion, une telle décomposition dans l'espace des formes s'exprime généralement par la donnée d'un radical (par ex. *parl*) et d'un affixe flexionnel (*ons*), correspondant respectivement à un lexème (`PARLER`) et à un ensemble de descripteurs (`pluriel+2+présent`). À l'opposé, dans la vision paradigmatique présentée, il est uniquement question de proportions formelles opérant indépendamment chacune à leur niveau ; nous dirons qu'elles sont *intra-niveau*.

Prenons un exemple. À l'aide de l'algorithme APPA, il est possible d'analyser automatiquement des formes (leur associer un lexème et des descripteurs) ; le processus inverse (générer une forme à partir d'un lemme et d'un ensemble de descripteurs) est également envisageable. Ces deux processus sont illustrés sur les figures 3.5 et 3.6.

nous parlons;`parler;pluriel+2+présent`	:	*nous marchons*;`marcher;pluriel+2+présent`
	::	
ils parlèrent;`parler;pluriel+3+passé`	:	*ils marchèrent*;`?;?`

FIG. 3.5 – Analyse de *marchèrent*

nous parlons;`parler;pluriel+2+présent`	:	*nous marchons*;`marcher;pluriel+2+présent`
	::	
ils parlèrent;`parler;pluriel+3+passé`	:	*?*;`marcher:pluriel+3+passé`

FIG. 3.6 – Génération de *marchèrent*

Cette exemple met clairement en évidence le traitement séparé des niveaux de représentation ; en effet, l'analyse de *marchèrent* implique d'une part la proportion formelle

nous parlons : *nous marchons* :: *ils parlèrent* : *ils marchèrent*,

et d'autre part l'équation (formelle)

`parler:pluriel+2+présent` : `marcher:pluriel+2+présent` :: `parler:pluriel+3+passé` : `?:?`.

Si l'on sait identifier les proportions formelles impliquées, alors il est possible d'éviter pour la description morphologique l'introduction de la notion de morphème comme unité minimale et de se passer de l'hypothèse de compositionnalité, qui postule que les propriétés d'une entité linguistique sont déductibles de sa

structure et des propriétés de ses constituants. Pour une discussion plus en profondeur à ce sujet, voir Pirrelli & Yvon (1999).

Signalons enfin que dans le cas de l'apprentissage de séquences, les mêmes hypothèses (décompositions en unités minimales et correspondance inter-niveaux) apparaissent implicitement dans les approches statistiques ou reposant sur une reformulation en un problème de classification. En effet, elles font apparaître la décomposition des objets x et y en séquences de symboles $x_1 \dots x_i$ et $y_1 \dots y_i$ (unités minimales), avec une correspondance inter-niveaux exprimée par des quantités telles que $\mathbb{P}(x_i|y_i)$ ou $\mathbb{P}(y_i|x_i)$.

cf. Sec. 3.1.2, p. 45

Paradigmes constructionnels La *construction* désigne le procédé par lequel des lexèmes nouveaux sont créés à partir de lexèmes existants. On dira par exemple que le nom CHANTEUR est créé à partir du verbe CHANTER par construction. Alors que la flexion n'opère aucune modification sur le sens et la catégorie grammaticale, ce n'est pas nécessairement le cas de la construction. Par exemple, CHANTEUR est un nom alors que CHANTER est un verbe.

La construction implique également des proportions analogiques formelles, telles que :

chanter : *chanteur* :: *lutter* : *lutteur*,
généreux : *générosité* :: *curieux* : *curiosité*,
agréable : *agréablement* :: *arbitraire* : *arbitrairement*.

En revanche, contrairement à la flexion, la construction n'est pas systématique ; les paradigmes sous-jacents à ces proportions sont donc identifiés moins clairement et peuvent être incomplets. Ce processus, sans être systématique, est toutefois très productif [23] ; des mots tels que *surfonctionné*, *jumenterie*, *chariatisation*, *saisonnalité*, *lichénisés* sont

> immédiatement compréhensibles pour tout locuteur du français. Ce sont des mots potentiels du français.
>
> Fradin (2003, p. 216)

On retrouve ici le lien avec l'extension analogique d'un ensemble, dont les éléments sont en puissance dans celui-ci.

Complétons maintenant la discussion à propos des unités minimales et de la correspondance inter-niveaux, dans le cas de la morphologie constructionnelle, à l'aide de la remarque suivante de Saussure, déjà évoqué p. 58.

> Un mot que j'improvise, comme *in-décor-able*, existe déjà en puissance dans la langue ; on retrouve tous ses éléments dans les syntagmes tels que *décor-er : décor-ation*, *pardonn-able : mani-able*, *in-connu : in-sensé*, etc., et sa réalisation dans la parole est un fait insignifiant en comparaison de la possibilité de le former.
>
> de Saussure (1916, p. 227)

23. Ces créations ne posent généralement pas de problèmes, sauf peut-être d'ordre esthétique : « Apprenabilité : néologisme assez laid que j'emploierai pour désigner "le fait d'être appris". » Denis (2000, p. 5).

Cette description pourrait tout d'abord laisser penser que Saussure fait référence à une décomposition d'*indécorable* en unités (*in+décor+able*), unités se retrouvant dans d'autres formes. Il n'en est rien.

> Pour former *indécorable*, nul besoin d'en extraire les éléments (*in-décor-able*) ; il suffit de prendre l'ensemble et de le placer dans l'équation :
>
> *pardonner : impardonnable, etc., = décorer : x.*
> *x = indécorable.*
>
> de Saussure (1916, p. 228-229)

Ainsi, dans cet exemple, seules comptent la proportion dans l'espace des formes et celle dans l'espace des sens [24].

3.4.3 Paradigmes syntaxiques

Le cas des « paradigmes syntaxiques » est traité par Matthews (1981, chap. 12, Syntactic Paradigms). Les paradigmes qu'il considère regroupent par exemple les phrases :

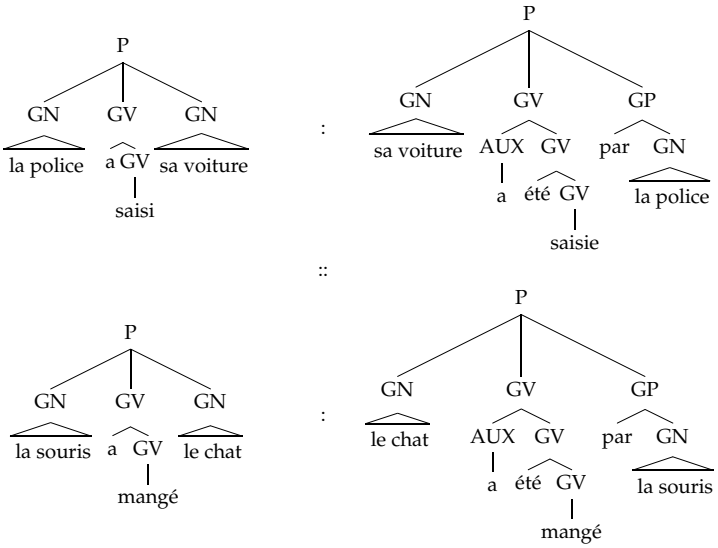
1. La police a saisi sa voiture.
2. Sa voiture a été saisie par la police.
3. La police a-t-elle saisi sa voiture ?
4. Sa voiture a-t-elle été saisie par la police ?

Celles-ci partagent un noyau commun, mais présentent des différences pouvant être décrites par des propriétés telles que la *passivation* et l'*interrogation*. De même, il est possible de considérer d'autres propriétés, comme la *négation*, donnant lieu aux phrases suivantes :

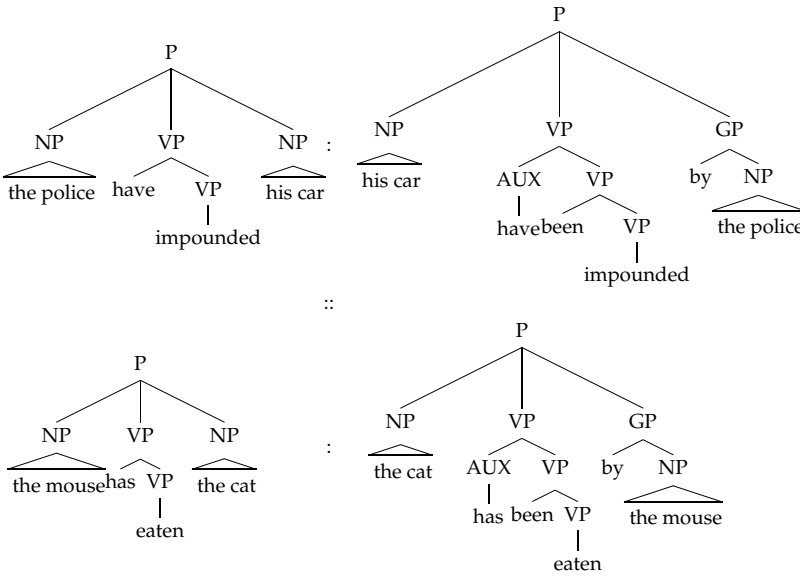
5. La police n'a pas saisi sa voiture.
6. Sa voiture n'a pas été saisie par la police.
7. La police n'a-t-elle pas saisi sa voiture ?
8. Sa voiture n'a-t-elle pas été saisie par la police ?

Chaque phrase véhicule de la sorte un ensemble de « traits » syntaxiques. Il est ici possible de mettre en relation les cases de différents paradigmes de façon à obtenir des proportions formelles (sur les représentations arborescentes des phrases), comme dans :

24. Afin d'éclairer quelque peu le « mystère » autour de cette notion de proportion analogique, indiquons que les mécanismes sous-jacents à celle-ci impliqueront des décompositions en facteurs qui alternent. Ces facteurs ne sont pas des morphèmes car leur unique rôle est de permettre l'identification des proportions, et aucun sens ne leur est associé. Les questions relatives à ce sujet trouveront leurs réponses dans le chapitre suivant.



ou



3.4.4 Paradigmes sémantiques

Les proportions que l'on considère sont formelles, c'est-à-dire identifiables uniquement à partir de la forme des objets qui la composent. Cela signifie en particulier que nous ne pourrons traiter directement des analogies sémantiques telles que « *la poule* est au *poussin* ce que *la jument* est au *poulain* ».

En revanche, si l'on dispose d'une représentation sémantique adéquate des termes composant cette proportion, notre modèle reste applicable. Les représentations sémantiques sont rencontrées sous diverses formes : structures de traits, prédicats d'une logique du premier ordre, ou encore nœud dans un graphe. En utilisant des prédicats sur une logique du premier ordre, la proportion précédente peut être exprimée sous la forme suivante.

$$\begin{array}{ccc} poule(X) & : & petit(Y,X) \wedge poule(X) \\ & :: & \\ jument(X) & : & petit(Y,X) \wedge jument(X) \end{array}$$

Dans le cas des structures de traits, nous avons :

$$\text{poule} = \begin{bmatrix} type & : gallinacé \\ genre & : femelle \\ age & : adulte \end{bmatrix} \quad : \quad \text{poussin} = \begin{bmatrix} type & : gallinacé \\ age & : petit \end{bmatrix}$$

$$::$$

$$\text{jument} = \begin{bmatrix} type & : cheval \\ genre & : femelle \\ age & : adulte \end{bmatrix} \quad : \quad \text{poulain} = \begin{bmatrix} type & : cheval \\ age & : petit \end{bmatrix}$$

Des proportions analogiques sont identifiables formellement à partir de ces représentations ; elles permettent donc de rendre compte des proportions analogiques entre les entités sous-jacentes. Ici, l'élément important à retenir est que l'identification des proportions s'opère uniquement à partir de représentations et de leurs relations structurelles ; elle ne fait pas appel à des connaissances extérieures.

3.5 Conclusion

Dans ce chapitre, nous avons étudié le problème de l'apprentissage des changements de niveaux de représentations linguistiques. Ce problème présente des spécificités qui ne sont pas traitées usuellement par les méthodes d'apprentissage automatique supervisé : symétrie entrée/sortie, objets structurés en entrée et en sortie, solutions non contenues dans la base d'apprentissage. Pour traiter ces problèmes, la méthode d'apprentissage présentée, APPA, exploite la structure paradigmatique des données linguistiques, en recherchant les proportions analogiques contenues dans celles-ci. Elle repose sur l'hypothèse qu'une proportion analogique formelle identifiée dans un niveau de représentation peut être le signe d'une proportion plus large (vérifiée pour d'autres niveaux de représentation). En outre, il est possible de formaliser son biais d'apprentissage à l'aide de la notion d'extension analogique.

Cette méthode est très souple car elle ne fait pas d'hypothèse sur la nature des objets traités : son « noyau » d'apprentissage est donc générique. En revanche, elle fait appel à des procédures (identification des proportions dans l'espace d'entrée et résolution d'équations analogiques dans l'espace de sortie), qui elles dépendent des objets considérés. Dans la suite, nous présentons un modèle formel permettant de caractériser la notion de proportion analogique pour des représentations courantes d'objets linguistiques, telles que les chaînes de symboles, les arbres, les structures de traits et les langages finis ; nous étudions également les procédures associées.

CHAPITRE

4 Modèles formels de proportions analogiques

Mieux vaut être belle et rebelle que moche et remoche.

— Richard Gotainer

Boire un café est possible.
Manger un restaurant n'est pas possible[1].

— Le chat à Malibu *(Philippe Geluck)*

Une civilisation sans la science, c'est aussi absurde qu'un poisson sans bicyclette.

— Pierre Desproges

Mon maître est le fils de Zeus, c'est un demi-dieu.
Moi, je suis le fils du chien de Zeus, je suis un demi-chien.

— Socrate le demi-chien *(Sfar & Blain)*

Sommaire du chapitre

1. Variante du proverbe « *Il est plus facile de boire un café que de manger un restaurant* ».

DANS le chapitre précédent, nous avons étudié les caractéristiques d'une méthode d'apprentissage par analogie, qui repose sur la notion de proportion et qui ne fait pas d'hypothèse sur la nature des objets manipulés. Deux procédures lui sont nécessaires. La première recherche les exemples de la base d'apprentissage formant avec l'objet à analyser une proportion analogique dans l'espace d'entrée. Cette procédure attend un objet $i(t)$ et fournit des triplets $(i(x), i(y), i(z))$ tels que $i(x) : i(y) :: i(z) : i(t)$. La deuxième résout des équations analogiques de la forme $o(x) : o(y) :: o(z) :?$ dans l'espace de sortie. De telles procédures doivent être en mesure de manipuler les types d'objets que l'on souhaite traiter, à savoir les structures habituellement utilisées en TAL : chaînes de symboles, arbres, structures de traits, langages finis, etc.

Néanmoins, avant de pouvoir proposer de tels mécanismes, il s'agit de savoir donner un sens à la notion de proportion analogique. Jusqu'ici, nous avons parlé de proportions sans pour autant définir cette notion ; nous nous sommes intégralement reposé sur une caractérisation informelle :

Quatre objets a, b, c, d forment une proportion analogique si et seulement si
« *a* est à *b* ce que *c* est à *d* ».

Il nous faut désormais répondre aux questions :

- quelles sont les relation entretenues par les objets formant une telle proportion ?
- quelles contraintes doivent-ils vérifier ?
- pourquoi la relation *marchons* : *marcher* :: *parlons* : *parler* est-elle qualifiable de proportion analogique alors que *courir* : *immeuble* :: *bateau* : *fourchette* ne l'est pas ? etc.

En d'autres termes, il faut donner un sens au rapport unissant ces objets.

> L'analogie est d'ordre grammatical : elle suppose la conscience et la compréhension d'un rapport unissant les formes entre elles.
>
> de Saussure (1916, p. 226)

Ce chapitre est consacré d'une part à la caractérisation formelle de la notion de proportion analogique et d'autre part à la proposition de procédures permettant de résoudre les tâches évoquées plus haut (recherche de triplets analogues et résolution d'équation). L'approche que nous adoptons est algébrique et structurelle : les objets que nous considérons sont des éléments d'une structure, c'est-à-dire un ensemble muni d'opérations. Dans ce cadre, la définition que nous donnons à la notion de proportion analogique est totalement liée à la structure algébrique sous-jacente aux objets considérés. L'analogie :

$$2 : 8 :: 3 : 12$$

n'aura un sens que parce que l'on aura préalablement défini une notion de proportion sur l'espace $(\mathbb{N}, \times)$, semigroupe abélien dont les quatre termes font partie. Nous travaillerons donc sur des *représentations* ; le choix d'une représentation pour les objets considérés devra être clairement fixé a priori [2].

2. Nous discuterons les implications d'une telle hypothèse dans la section 4.2.4. Nous y étudie-

L'ensemble des mots[3], chaînes de symboles d'un alphabet Σ, forme un monoïde libre $\Sigma^\star$. Un monoïde libre est une structure algébrique, très largement étudiée, aux fondements de l'informatique théorique et de la théorie des langages formels ; c'est également un objet d'étude privilégiée de la linguistique computationnelle. Le besoin de définir la notion de proportion analogique s'est donc naturellement présenté dans un premier temps pour cette structure, donnant lieu au modèle proposé par Yvon (2003). Un des objectifs de notre travail consiste à **étendre ce modèle de proportions entre mots de façon à être en mesure d'appliquer l'algorithme APPA à d'autres types de représentations linguistiques**.

Le modèle de proportion entre mots en question constitue le point de départ de nos considérations ; nous l'étudions et le complétons dans la section 4.1. Nous montrons qu'il peut s'appliquer en réalité à des structures algébriques plus générales, telles que les semigroupes et même les magmas[4] (cf. section 4.2). Cette généralisation fournit un cadre algébrique unifié et cohérent. Par exemple, elle permet, à l'aide du même modèle, de couvrir le cas des monoïdes libres, des semigroupes abéliens, des treillis et des groupes. Il est ensuite possible d'instancier naturellement ce modèle générique de manière à obtenir une définition de la notion de proportion analogique entre représentations courantes d'objets linguistiques. Pour certaines des structures étudiées, nous discutons également des possibilités d'implantation des algorithmes visés, à savoir la recherche de triplets analogues et la résolution d'équations.

Exemples

Nous présentons dans ce chapitre des modèles formels de proportions analogiques. L'objectif de cette modélisation ne se limite pas à un jeu intellectuel de manipulation de structures algébriques. La notion de proportion analogique correspond à un phénomène, caractérisé par une capacité à produire et à discerner de telles proportions, notamment dans un contexte langagier. C'est par ailleurs sur cette capacité que nous nous sommes reposé, puisqu'aucune définition formelle n'a jusqu'à maintenant été fournie au lecteur. C'est ce phénomène qu'il faut savoir expliquer précisément.

Afin de caractériser cette capacité, nous nous reposons sur un certain nombre d'exemples que nous voulons modéliser, à savoir les proportions impliquées dans des paradigmes flexionnels, dérivationnels ou syntaxiques. Les définitions que nous allons proposer devront être en adéquation avec de tels exemples. Nous en présentons quelques-uns dans la suite, dans le cas des mots, des arbres, des structures de traits et des ensembles ; certains proviennent ou sont adaptés de Lepage (2003), Yvon (2003), Matthews (1981) et Fradin (2003).

---

rons également un modèle théorique permettant de s'en abstraire.

3. Dans ce chapitre, puisqu'il sera question de modèles formels, l'expression *mots* perdra toute connotation linguistique. Un mot sera synonyme de *chaîne (de symboles)*, et désignera un élément d'un monoïde libre $\Sigma^\star$ sur un alphabet Σ.

4. Si une structure algébrique est un ensemble muni d'une opération alors le magma est la structure la plus générale qui soit.

Proportions analogiques entre mots

décorer : indécorable :: démonter : indémontable
jouer : surjouer :: fonctionner : surfonctionner
oratorem : orator :: honorem : honor
wolf : wolves :: leaf : leaves
reader : unreadable :: doer : undoable
arsala : mursilim :: aslama : muslimim
arm : ärme :: spruch : sprüche

FIG. 4.1 – Exemples de proportions entre mots

Proportions analogiques entre arbres

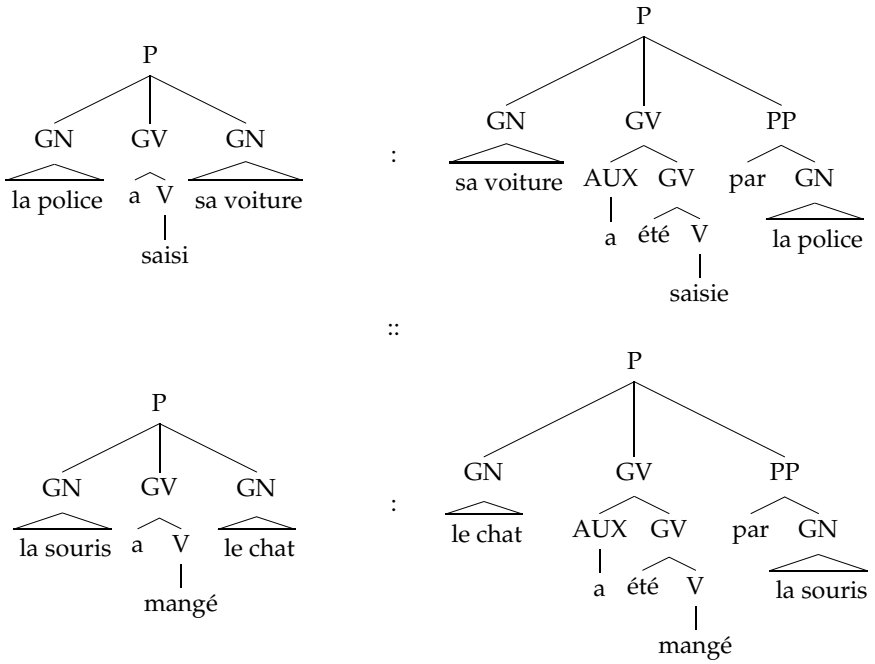


FIG. 4.2 – Exemple de proportion entre arbres

Proportions analogiques entre structures de traits

$$\left[\begin{array}{l} SUJET : \left[\begin{array}{l} [1]\textbf{accord} \\ PERSONNE : [\,\textbf{1re}\,] \\ NOMBRE : [\,\textbf{singulier}\,] \end{array}\right] \\ PREDICAT : [\,[1]\,] \end{array}\right] : \left[\begin{array}{l} SUJET : \left[\begin{array}{l} [1]\textbf{accord} \\ PERSONNE : [\,\textbf{3e}\,] \\ NOMBRE : [\,\textbf{singulier}\,] \end{array}\right] \\ PREDICAT : [\,[1]\,] \end{array}\right]$$

::

$$\left[\begin{array}{l} SUJET : \left[\begin{array}{l} [1]\textbf{accord} \\ PERSONNE : [\,\textbf{1re}\,] \\ NOMBRE : [\,\textbf{pluriel}\,] \end{array}\right] \\ PREDICAT : [\,[1]\,] \end{array}\right] : \left[\begin{array}{l} SUJET : \left[\begin{array}{l} [1]\textbf{accord} \\ PERSONNE : [\,\textbf{3e}\,] \\ NOMBRE : [\,\textbf{pluriel}\,] \end{array}\right] \\ PREDICAT : [\,[1]\,] \end{array}\right]$$

FIG. 4.3 – Exemple de proportion entre structures de traits

Proportions analogiques entre ensembles

$$\begin{array}{ccc} \mathrm{x} = \{\ /\mathrm{b}/, /\mathrm{p}/, /\mathrm{v}/, /\mathrm{f}/\ \} & : & \mathrm{y} = \{\ /\mathrm{b}/, /\mathrm{p}/, /\mathrm{d}/, /\mathrm{t}/\ \} \\ & :: & \\ \mathrm{z} = \{\ /\mathrm{z}/, /\mathrm{s}/, /\mathrm{v}/, /\mathrm{f}/\ \} & : & \mathrm{t} = \{\ /\mathrm{z}/, /\mathrm{s}/, /\mathrm{d}/, /\mathrm{t}/\ \} \end{array}$$

FIG. 4.4 – Exemple de proportion entre ensembles

Propriétés fondamentales

Parallèlement à ces exemples, un certain nombre de propriétés « universelles » doivent être prises en considération de manière à rester en cohérence avec l'acception traditionnelle de la proportion analogique en philosophie et en linguistique (Lepage, 2003).

Formellement, donner une définition de la proportion analogique sur un ensemble E revient à caractériser une relation quaternaire sur E, c'est-à-dire un sous-ensemble de quadruplets $A \subseteq E^4$. Plusieurs définitions pourront coexister pour un même ensemble E ; en cas d'ambiguïté, nous spécifierons la définition considérée par la notation $x : y \overset{A}{::} z : t$.

Cette relation doit tout d'abord présenter un certain nombre de propriétés de symétrie.

Condition 1 (Symétrie de la conformité). *Les « côtés » gauche et droit d'une proportion peuvent être permutés.*

$$x : y :: z : t \Rightarrow z : t :: x : y.$$

Condition 2 (Permutation des moyennes). *Les deux termes centraux d'une proportion peuvent être permutés.*

$$x : y :: z : t \Rightarrow x : z :: y : t.$$

De ces conditions, on peut tirer facilement cinq autres expressions équivalentes à $x : y :: z : t$ [5].

Inversion des rapports :	$y : x :: t : z$
Permutation des extrêmes :	$t : y :: z : x$
Symétrie de la lecture :	$t : z :: y : x$
	$y : t :: x : z$
	$z : x :: t : y$

En outre, on considère que les proportions de la forme $a : a :: b : b$ et $a : b :: a : b$ sont toujours vérifiées. Nous appellerons *proportions atomiques* [6] de telles proportions.

5. Toutes ces propriétés sont vérifiées pour un parallélogramme ; par exemple, $(\vec{x}, \vec{y}, \vec{z}, \vec{t})$ est un parallélogramme implique que $(\vec{z}, \vec{t}, \vec{x}, \vec{y})$ est un parallélogramme.

6. Le choix de l'adjectif atomique sera rendu clair dans la suite.

Condition 3 (Proportions atomiques). *A doit couvrir les* proportions atomiques.

$$\forall (a,b) \in E^2, a : b :: a : b \text{ et } a : a :: b : b.$$

Un axiome facultatif (appelé *déterminisme*), reposant sur la validité des proportions atomiques, assure que l'équation $x : x :: z :?$ a une seule solution.

Condition 4 (Déterminisme).

$$x : x :: z : t \Rightarrow t = z.$$

4.1 Proportions analogiques entre mots : un point de départ

4.1.1 Introduction

Dans cette section, nous reprenons la définition de l'analogie entre mots proposée initialement par Yvon (2003) et reformulée dans Yvon *et al.* (2004). Nous discutons quelques-unes de ses propriétés, et proposons quelques mesures de qualité à associer à une proportion. Avant de poursuivre, nous fixons un certain nombre de notations utiles pour la suite.

Notations

Soit Σ un ensemble fini de symboles, appelé *alphabet*. $\Sigma^\star$ désigne l'ensemble des séquences finies de symboles de Σ, appelées *mots* sur Σ. Muni de l'opération de *concaténation*, $\Sigma^\star$ est un *monoïde libre* dont l'*élément neutre* est le mot vide ϵ, noté également $1_{\Sigma^\star}$. Pour x et y dans $\Sigma^\star$, la concaténation de x et y est notée $x.y$ ou simplement xy. Nous notons $|x|$ la *longueur* de x ; par définition, on a $|\epsilon| = 0$. Pour $x \in \Sigma^\star$ et $i \leq |x|$, $x(i)$ désigne le i^{e} symbole de x.

Un ensemble (ordonné) d'indices est un ensemble d'entiers positifs $\{i_1, \ldots, i_n\}$ tel que $i_1 < i_2 < \cdots < i_n$. L'ensemble des ensembles d'indices est noté $\mathcal{J}$. Pour deux ensembles d'indices disjoints $I = \{i_1, \ldots, i_n\}$ et $J = \{j_1, \ldots, j_m\}$, $I + J$ désigne l'ensemble ordonné d'indices contenant les entiers de $I \cup J$. Les ensembles d'entiers consécutifs de i à j seront notés $[\![i, j]\!]$ et appelés intervalles. Si u est un mot, I_u représente l'intervalle $[\![1, |u|]\!]$. L'ensemble de tous les intervalles est noté $\mathcal{I}$.

Si $w = uzv$, u est un *préfixe* de w, v un *suffixe* de w, et z un *facteur* de w. Un mot v est un *sous-mot* de $w = w(1) \ldots w(n)$ si et seulement s'il existe un ensemble d'indices $S = \{i_1, \ldots, i_k\} \subseteq [\![1, n]\!]$ tel que $w(i_1) \ldots w(i_k) = v$. Une correspondance (non nécessairement unique) entre I_v et S, qui associe à chaque position de v une position de w, est notée $\phi_{w,v}$. Par définition, on a $\forall i \in I_v$, $v(i) = w(\phi_{w,v}(i))$. Réciproquement, tout ensemble d'indices I inclus dans $[\![1, n]\!]$ induit un sous mot de w, noté $w(I)$. La notion de sous-mot définit une relation représentée par le symbole $\in$.

Les facteurs sont des cas particuliers de sous-mots, induits par des intervalles. Les préfixes et les suffixes sont des cas particuliers de facteurs, induits respectivement par des ensembles d'indices de la forme $[\![1, i]\!]$ et $[\![i, |w|]\!]$. On notera[7] $x(i : j)$ le facteur $x([\![i, j]\!])$, $x(: i)$ le préfixe $x(1 : i)$, et $x(i :)$ le suffixe $x(i : |x|)$.

L'ensemble des préfixes d'un mot u est noté $\mathcal{P}\text{ref}(u) = \{v \mid \exists w \ t.q. \ u = vw\}$. Cette notion s'étend naturellement aux langages : $\mathcal{P}\text{ref}(L) = \bigcup_{u \in L} \mathcal{P}\text{ref}(u)$. Un langage $L \subseteq \Sigma^\star$ est dit *préfixiel* si et seulement si $\mathcal{P}\text{ref}(L) = L$. Le *quotient* (gauche) d'un langage L par un mot u est défini par $u^{-1}L = \{v | uv \in L\}$.

Une *factorisation* d'un mot w est une séquence $f_w = (w_1, w_2, \ldots, w_m)$ telle que $\forall i \in [\![1, m]\!]$, $w_i \in \Sigma^\star$ et $w_1 w_2 \ldots w_m = w$. On notera $f_w(i) = w_i$. On notera f_w une factorisation de w. Il est facile de vérifier que chaque w_i est un facteur de w.

Un automate fini A est caractérisé par la donnée de 5 éléments $\langle \Sigma, Q, I, F, E \rangle$, où Σ est un alphabet fini, Q un ensemble fini d'états, $I \subseteq Q$ un ensemble d'états initiaux, $F \subseteq Q$ un ensemble d'états finals, et $E \subseteq Q \times A \times Q$ un ensemble de transitions. Une suite de transitions de la forme $(q_0, a_1, q_1)(q_1, a_2, q_2) \ldots (q_{n-1}, a_n, q_n)$ et telle que $q_0 \in I$ et $q_n \in F$ est un calcul réussi. Le mot $a_1 a_2 \ldots a_n$ est dit accepté (ou reconnu) par A ; le langage $L(A)$ accepté par A est l'ensemble des mots qu'il accepte. Un transducteur T est un automate sur un produit de monoïdes libres $\Sigma_1^* \times \Sigma_2^*$. Les transducteurs étudiés seront étiquetés par des éléments de $(\Sigma_1 \cup \{1_{\Sigma_1^*}\}) \times (\Sigma_2 \cup \{1_{\Sigma_2^*}\})$. La première composante de ces couples s'appellera l'entrée et la deuxième la sortie. Dans la suite de ce travail, nous supposerons toujours $\Sigma_1 = \Sigma_2 = \Sigma$ et noterons $1_\Sigma = \epsilon$. Si T est un transducteur, $P_E(T)$ (resp. $P_S(T)$) désigne l'automate obtenu en prenant la projection de chaque étiquette sur sa première (resp. sa deuxième) composante ; $P_E(T)$ (resp. $P_S(T)$) sera l'automate d'entrée (resp. de sortie) sous-jacent à T. $L_E(T) = L(P_E(T))$ (resp. $L_S(T) = L(P_S(T))$) sera le langage d'entrée (resp. de sortie) de T.

4.1.2 Définition et propriétés

Définition

Le modèle proposé par Yvon (2003) donne lieu à la définition suivante de la proportion entre mots, présentée ici de manière légèrement différente.

Définition 4 (Proportion analogique (entre mots)). Pour $(x, y, z, t) \in \Sigma^{\star 4}$, on a $x : y :: z : t$ si et seulement s'il existe des factorisations $(f_x, f_y, f_z, f_t) \in ((\Sigma^\star)^d)^4$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}.$$

Par construction, on a $|f_x| = |f_y| = |f_z| = |f_z| = d$. Le plus petit d pour lequel de telles factorisations peuvent être trouvées est nommé le *degré* de la proportion. L'ensemble des factorisations vérifiant cette propriété sera noté $\mathcal{AF}(x : y :: z : t)$ ou simplement $\mathcal{AF}$.

7. Clin d'œil au langage PYTHON (`http://www.python.org`).

Cette définition contient deux ingrédients : une *décomposition* des mots en facteurs, et un *mécanisme d'alternance* qui oppose les facteurs deux-à-deux. Prenons le cas de la proportion *décorer* : *indécorable* :: *démonter* : *indémontable*. Pour rendre compte de cette proportion, les quatre termes impliqués sont factorisés de la manière suivante :

- x = *décorer* ; f_x = (*ϵ*, *décor*, *er*)
- y = *indécorable* ; f_y = (*in*, *décor*, *able*)
- z = *démonter* ; f_z = (*ϵ*, *démont*, *er*)
- t = *indémontable* ; f_t = (*in*, *démont*, *able*)

Ensuite, on peut vérifier que les facteurs alternent deux-à-deux, ce qui s'exprime par la condition

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}.$$

Une autre façon d'écrire cette alternance est

$$\forall i \in [\![1, d]\!], (f_x(i) = f_y(i) \text{ et } f_z(i) = f_t(i)) \text{ ou } (f_x(i) = f_z(i) \text{ et } f_y(i) = f_t(i)).$$

Cette propriété bien est vérifiée pour les trois quadruplets de facteurs :

(ϵ, in, ϵ, in), *(décor, décor, démont, démont)*, et *(er, able, er, able)*.

Chaque quadruplet de facteurs forme une proportion analogique atomique, c'est-à-dire une proportion de la forme $a : a :: b : b$ ou $a : b :: a : b$. Chaque factorisation comprend 3 facteurs, le degré d de la proportion est donc égal à 3.

Cette définition correspond à l'intuition qu'une proportion analogique implique des mots partageant des facteurs communs qui alternent. Elle est pertinente pour plusieurs raisons. Tout d'abord, elle permet de modéliser correctement les exemples présentés dans la section 4. Ainsi, pour

arsala : *mursilim* :: *aslama* : *muslimim*,

on a les factorisations et les alternances suivantes.

a	rs	a	l	a
mu	rs	i	l	im
a	sl	a	m	a
mu	sl	i	m	im

Ensuite, il est facilement vérifiable qu'elle répond aux conditions imposées dans l'introduction de ce chapitre. Enfin, il est possible d'en dériver des constructions à base d'automates et de transducteurs finis de manière à calculer les proportions (cf. Yvon (2003) ; Yvon *et al.* (2004) et section 4.1.3). Dans la suite, c'est cette définition (la définition 4) qui servira de base à la généralisation. Les travaux apparentés de Lepage (1998) et Miclet *et al.* (2005) sont discutés dans l'annexe A.

Propriétés

De la définition proposée, il est possible de dériver un certain nombre de propriétés.

Proposition 1. Si $x : y :: z : t$ et $x' : y' :: z' : t'$, alors $xx' : yy' :: zz' : tt'$.

Démonstration. Direct par concaténation des factorisations. □

Une *factorisation simple* f_x est telle que $\forall i \in I_{f_x}, |f_x(i)| \in \{0, 1\}$.

Proposition 2. Si $x : y :: z : t$, alors il existe un quadruplet de factorisations simples $(f_x, f_y, f_z, f_t) \in \Sigma^{\star d}$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\} \text{ et } f_x(i)f_y(i)f_z(i)f_t(i) \neq \epsilon.$$

Démonstration. Pour tout mot x et tout $n \geq |x|$, on peut construire une factorisation simple f_x de taille n, en posant

$$f_x(i) = \begin{cases} x(i) & \text{pour } i \leq |x|, \\ \epsilon & \text{pour } i > |x|. \end{cases}$$

On note $s_d(x, n)$ une telle factorisation simple.

Par hypothèse, il existe des factorisations $(f_x, f_y, f_z, f_t) \in \Sigma^{\star d}$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}.$$

Soit $\Lambda = \{x, y, z, t\}$. Pour tout $\alpha \in \Lambda$, et tout $i \in [\![1, d]\!]$, $f'_{\alpha,i}$ est la factorisation simple construite à partir du mot $f_\alpha(i)$ selon la méthode décrite ci-dessus, de taille $\max_{\alpha\in\Lambda} |f_\alpha(i)|$, i.e. $f'_{\alpha,i} = s_d(f_\alpha(i), \max_{\alpha\in\Lambda} |f_\alpha(i)|)$.

Pour $\alpha \in \Lambda$, en considérant la factorisation simple obtenue en concaténant les factorisations simples $f'_{\alpha,i}$, on trouve le résultat souhaité. □

Cela signifie que pour rendre compte d'une proportion, il est possible de considérer uniquement des factorisations dont les éléments sont des lettres de Σ ou ϵ. Pour la proportion *arsala* : *mursilim* :: *aslama* : *muslimim*, cela donne :

a	ϵ	r	s	a	l	a	ϵ
m	u	r	s	i	l	i	m
a	ϵ	s	l	a	m	a	ϵ
m	u	s	l	i	m	i	m

Proposition 3. Si $x : y :: z : t$, alors il existe un quadruplet de factorisations simples $(f_x, f_y, f_z, f_t) \in \Sigma^{\star d}$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\} \text{ et } |f_x(i)| + |f_t(i)| = 1.$$

Démonstration. Pour démontrer le résultat, il suffit de considérer les factorisations obtenues en « séparant » tout quadruplet de facteurs $(f_x(i), f_y(i), f_z(i), f_t(i))$ tel que $f_x(i) = f_y(i) = f_z(i) = f_t(i) = 1$ en deux quadruplets de facteurs. Par exemple, en supposant $f_x(i) = f_y(i)$ (par symétrie), on peut considérer les quadruplets $(f_x(i), f_y(i), \epsilon, \epsilon)$ et $(\epsilon, \epsilon, f_z(i), f_t(i))$. □

Pour la proportion *arsala* : *mursilim* :: *aslama* : *muslimim*, on obtient :

ϵ	ϵ	a	ϵ	ϵ	r	s	ϵ	a	ϵ	l	ϵ	ϵ	a
m	u	ϵ	ϵ	ϵ	r	s	i	ϵ	ϵ	l	i	m	ϵ
ϵ	ϵ	a	s	l	ϵ	ϵ	ϵ	a	m	ϵ	ϵ	ϵ	a
m	u	ϵ	s	l	ϵ	ϵ	i	ϵ	m	ϵ	i	m	ϵ

Proposition 4.

$$x : y :: z : t \Leftrightarrow \begin{cases} x(2:) : y(2:) :: z : t & \text{et} \quad x(1) = y(1) \quad \text{ou} \\ x(2:) : y :: z(2:) : t & \text{et} \quad x(1) = z(1) \quad \text{ou} \\ x : y(2:) :: z : t(2:) & \text{et} \quad t(1) = y(1) \quad \text{ou} \\ x : y :: z(2:) : t(2:) & \text{et} \quad t(1) = z(1) \end{cases}$$

Démonstration. $\Leftarrow$. Direct par utilisation de la proposition 1.

$\Rightarrow$. Direct par utilisation de la proposition 3. □

cf. Sec. 3.4, p. 65

Remarque Dans le chapitre précédent, nous avions évoqué le traitement séparé des niveaux de représentations et l'absence de correspondance entre les unités minimales d'un niveau et celles d'un autre niveau ; l'existence de ces unités minimales n'est par ailleurs pas nécessaire au fonctionnement du modèle. On retrouve pourtant, dans la notion de proportion analogique présentée, une notion de décomposition des entités en facteurs « plus petits » ; cela mérite quelques éclaircissements.

Il est important de ne pas confondre la décomposition impliquée dans la définition proposée de la proportion analogique, avec une approche concaténative, par exemple de la morphologie. En effet, les facteurs intervenant dans une proportion n'ont pas d'existence propre et il n'est jamais nécessaire de les exhiber explicitement. Leur unique rôle est de permettre de déterminer si quatre termes étudiés forment une proportion. Une fois la proportion reconnue comme telle, ces facteurs deviennent inutiles ; ils sont construits dynamiquement et leur existence est éphémère. En outre, aucun sens ne leur est associé et aucune correspondance n'est établie entre ces facteurs et d'autres facteurs d'un niveau de représentation différent. Par exemple, dans la proportion (valide) *sing* : *sang* :: *ring* : *rang*, les facteurs impliqués sont *s*, *r*, *ing*, et *ang*. Il n'est pas besoin de donner un sens aux « morphèmes » *s* ou *r* pour identifier cette proportion et l'exploiter dans un contexte inférentiel.

4.1.3 Méthodes de calcul

Des définitions proposées, il est possible de dériver un certain nombre d'algorithmes, permettant de vérifier une proportion et de résoudre une équation. Deux types d'algorithmes sont présentés, les uns reposant sur la construction explicite d'automates, les autres sur des méthodes de programmation dynamique.

Sous-mots complémentaires et produit de mélange

Les constructions à base d'automates finis nécessitent l'introduction des notions de mots complémentaires et de produit de mélange.

Sous-mots complémentaires Si v est un sous-mot de w, l'*ensemble des complémentaires* de v par rapport à à w, noté $w\backslash v$ est l'ensemble des sous-mots de w obtenus en supprimant de w, de la gauche vers la droite, les symboles de v. Par exemple, *eeaie* est un sous-mot complémentaire de *xmplr* relativement à *exemplaire*. Cette notion est formalisée par la définition qui suit.

Définition 5 (Sous-mots complémentaires). L'ensemble des sous-mots complémentaires de v par rapport à w est défini par :

$$w\backslash v = \{u \mid \exists I \subseteq I_w, w(I) = u, w(I_w \setminus I) = v\}.$$

En particulier, si v n'est pas un sous-mot de w, $w\backslash v$ est vide.

L'ensemble complémentaire de v relativement à w est un langage rationnel : c'est le langage de sortie du transducteur fini T_w (cf. figure 4.5) pour l'entrée v.

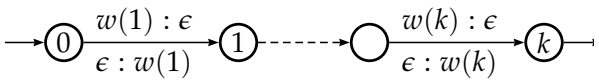


FIG. 4.5 – Le transducteur T_w calculant les ensembles complémentaires relativement à w.

À chaque mot w de $\Sigma^\star$ est associée une relation binaire sur $\Sigma^\star \times \Sigma^\star$, notée $\backslash_w$, et définie par :

Définition 6 (Complémentarité). $u\backslash_w v$ si et seulement si $u \in w\backslash v$.

Pour tout mot w de $\Sigma^\star$, il est aisé de construire un automate fini A_w qui reconnaît uniquement w en établissant une bijection entre les états de A_w et les préfixes de w. En ajoutant à chaque transition de A_w une transition spontanée, on dérive un automate S_w qui reconnaît exactement les sous-mots de w. En transformant S_w en un transducteur T_w tel que chaque transition de S_w étiquetée par w_i produit en sortie ϵ et chaque transition ϵ produit w_i, on obtient finalement une machine calculant la relation de complémentarité par rapport à w ; cette relation est donc rationnelle. Ces constructions sont illustrées sur la figure 4.6.

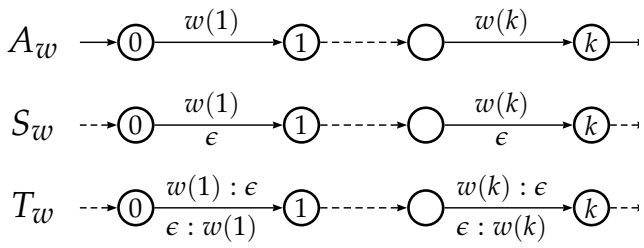


FIG. 4.6 – Automates et transducteurs pour w, ses sous-mots et la relation de complémentarité

On notera, pour finir, que les notions de sous-mots complémentaires et de relation de complémentarité s'étendent sans difficulté à des langages rationnels quelconques.

Produit de mélange Le produit de mélange de u et v (noté $u \bullet v$) contient tous les mots formés des symboles de u et de v, avec la contrainte que si a précède b dans u ou v, alors cet ordre est respecté dans $u \bullet v$. Par exemple, si l'on prend $u = abc$ et $v = def$, alors les mots $abcdef$, $abdefc$, $adbecf$ sont dans $u \bullet v$; ce n'est pas le cas de $abefcd$, dans lequel d suit e, alors qu'il devrait le précéder.

Le produit de mélange est défini par exemple par Sakarovitch (2003) comme suit.

Définition 7 (Produit de mélange). Si u et v sont deux mots de $\Sigma^\star$, leur mélange $u \bullet v$ est le langage :

$$u \bullet v = \{u_1 v_1 u_2 v_2 \dots u_n v_n, \text{ avec } u_i, v_i \in \Sigma^\star, u_1 \dots u_n = u, v_1 \dots v_n = v\}$$

Une autre façon d'écrire le produit de mélange est :

$$u \bullet v = \{w \in \Sigma^\star | \ \exists I \subseteq I_w, w(I) = u, w(I_w \setminus I) = v\}.$$

Cette opération s'étend naturellement aux langages. Pour les langages K et L :

$$K \bullet L = \bigcup_{x \in K, y \in L} x \bullet y$$

Il est connu que cette opération est une opération rationnelle, et que le mélange de deux mots est calculé en effectuant le produit (de mélange) des automates reconnaissant ces deux mots. Formellement, si K et L sont deux langages rationnels reconnus respectivement par $A_K = \langle \Sigma, Q_K, I_K, F_K, E_K \rangle$ et $A_L = \langle \Sigma, Q_L, I_L, F_L, E_L \rangle$, l'automate mélange reconnaissant $K \bullet L$ est décrit par la donnée des éléments $\langle \Sigma, Q_K \times Q_L, I_K \times I_L, F_K \times F_L, G \rangle$, avec $((p_K, p_L), a, (q_K, q_L)) \in G$ si et seulement si ou bien $(p_L, a, q_L) \in E_L$ et $p_K = q_K \in Q_K$ ou bien $(p_K, a, q_K) \in E_K$ et $p_L = q_L \in Q_L$ (cf. figure 4.7).

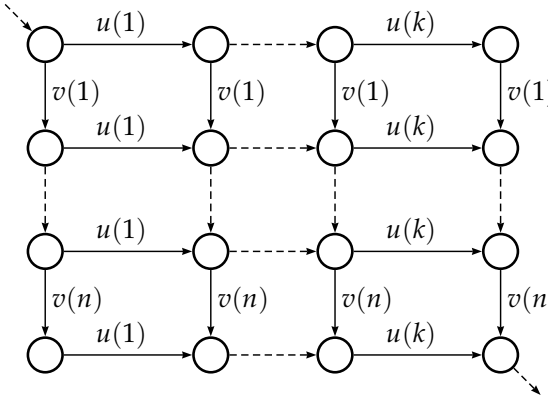


FIG. 4.7 – Automate mélange de u et v

Les notions de sous-mots complémentaires et de produit de mélange sont reliées par la relation suivante, conséquence directe des définitions :

$$w \in u \bullet v \Leftrightarrow u \in w \backslash v.$$

Elle énonce que u et v sont en relation de complémentaire par rapport à w si et seulement si w appartient au produit de mélange de ces deux mots.

Vérification de proportions et résolution d'équations analogiques Les notions de mots complémentaires et de produit de mélange fournissent une caractérisation alternative de la proportion analogique entre mots, fondée sur la propriété suivante.

Proposition 5.

$$\forall x, y, z, t \in \Sigma^\star, x : y :: z : t \Leftrightarrow x \bullet t \cap y \bullet z \neq \emptyset$$

Une proportion analogique est vérifiée si les symboles de x et t apparaissent également dans y et z, dans le même ordre.

Démonstration. $\Rightarrow$. Pour des factorisations (f_x, f_y, f_z, f_z) de taille d, il suffit de considérer le mot $w = f_x(1)f_t(1)\ldots f_x(d)f_t(d) \in x \bullet t$. Puisque

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\},$$

on a $w \in y \bullet z$, et $w \in x \bullet t \cap y \bullet z$.

$\Leftarrow$. Puisque $w \in x \bullet t \cap y \bullet z$, $\exists I_1, I_2 \subseteq I_w$ tels que

$$w(I_1) = x, w(I_w \setminus I_1) = t, w(I_2) = y, w(I_w \setminus I_2) = z.$$

Pour $i \in I_w$, on pose

$$(f_x(i), f_t(i)) = \begin{cases} (w(i), \epsilon) & \text{si } i \in I_1, \\ (\epsilon, w(i)) & \text{sinon,} \end{cases} \text{ et } (f_y(i), f_z(i)) = \begin{cases} (w(i), \epsilon) & \text{si } i \in I_2, \\ (\epsilon, w(i)) & \text{sinon.} \end{cases}$$

Pour $\alpha \in \{x, y, z, t\}$, f_α est une factorisation de α, de taille $|w|$, et on a bien :

$$\forall i \in I_w, (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\},$$

d'où le résultat recherché. □

Ce résultat a pour corollaire direct :

Proposition 6. t est une solution de l'équation analogique $x : y :: z :?$ si et seulement si

$$t \in (y \bullet z) \backslash x.$$

L'ensemble des solutions d'une équation analogique $x : y :: z :?$ est donc un ensemble rationnel, calculable par un transducteur fini.

Ces résultats nous fournissent deux constructions, une pour la vérification de proportions (cf. algorithme 2) et une pour la résolution d'équations (cf. algorithme 3) .

entrée : quatre mots x, y, z, t
sortie : un booléen, égal à $x : y :: z : t$
begin
  $xt \longleftarrow$ mélange(A_x, A_t)
  $yz \longleftarrow$ mélange(A_y, A_z)
  $inter \longleftarrow$ produit(xt, yz)
end
return est_vide($inter$)

Algorithme 2 : vérifie_proportion

entrée : trois mots x, y, z
sortie : un automate reconnaissant l'ensemble des solutions $S(x : y :: z :?)$
begin
  $yz \longleftarrow$ mélange(A_y, A_z)
  $s \longleftarrow$ complémentaire(yz, A_x)
end
return s

Algorithme 3 : résout_équation

Le langage représenté par un automate est non-vide si et seulement s'il possède un état final accessible. L'automate $M(u, v)$, reconnaissant le produit de mélange de deux mots u, v, contient un unique état final accessible. Lorsque l'on construit un automate reconnaissant l'intersection des langages reconnus par deux automates, il est possible de ne considérer que les états accessibles de l'automate produit (cf. bibliothèque VAUCANSON (Lombardy *et al.*, 2004)). Vérifier que le langage reconnu par l'automate résultant de cette construction est non-vide se résume à observer s'il contient un état final ou non, opération s'opérant à coût constant. La

complexité de cette algorithme est donc de manière immédiate en $O(|x| \times |y| \times |z| \times |t|)$. Pour l'algorithme de résolution, la complexité est en $O(|x| \times |y| \times |z|)$; toutefois, l'ensemble des solutions est présenté sous la forme d'un automate fortement non-déterministe (dû à l'opération de complémentarité). Dans certains cas, cette représentation suffit (par ex. lorsqu'il s'agit de vérifier qu'un mot appartient à l'ensemble des solutions). Dans d'autres, une énumération explicite des solutions est nécessaire, conduisant à la déterminisation et à l'exploration de celui-ci. Voir l'annexe B pour plus de détails concernant ces questions de complexité.

Algorithmes tabulaires

Il est possible d'exploiter la proposition 4 de manière à effectuer les mêmes calculs à l'aide d'un algorithme de type programmation dynamique. Pour vérifier une proportion, on exploite l'invariant suivant, directement obtenu à partir de la proposition 4, où $ana(i,j,k,l)$ désigne $x(:i) : y(:j) :: z(:k) : t(:l)$.

$$ana(i,j,k,l) = \begin{cases} ana(i-1,j-1,k,l) & \text{et} \quad x(i){=}y(j) & \text{ou} \\ ana(i-1,j,k-1,l) & \text{et} \quad x(i){=}z(k) & \text{ou} \\ ana(i,j-1,k,l-1) & \text{et} \quad t(l){=}y(j) & \text{ou} \\ ana(i,j,k-1,l-1) & \text{et} \quad t(l){=}z(k). \end{cases}$$

On en déduit les algorithmes 4 et 5.

entrée : quatre mots x, y, z, t
sortie : un booléen, égal à $x : y :: z : t$
begin
  $ana(i,j,k,l) \leftarrow$ `faux` si $(i<0) \parallel (j<0) \parallel (k<0) \parallel (l<0)$
  for $i = 0; i \leq |x|$ **do**
    for $j = 0; j \leq |y|$ **do**
      for $k = 0; k \leq |z|$ **do**
        for $l = 0; l \leq |t|$ **do**
          if $i = j = k = l = 0$ **then**
            $ana(i,j,k,l) \leftarrow$ `vrai`
          else
            $ana(i,j,k,l) \leftarrow ana(i-1,j-1,k,l) \& x(i) = y(j)$
            $\parallel ana(i-1,j,k-1,l)\ \&\ x(i) = z(k)$
            $\parallel ana(i,j-1,k,l-1)\ \&\ t(l) = y(j)$
            $\parallel ana(i,j,k-1,l-1)\ \&\ t(l) = z(k)$
          end
        end
      end
    end
  end
end
return $ana(|x|)(|y|)(|y|)(|z|)$

Algorithme 4 : vérifie_proportion_tab

entrée : trois mots x, y, z
sortie : L'ensemble des solutions $S(x : y :: z :?)$
begin
  $s(i,j,k) \leftarrow \emptyset$ si $(i < 0) \parallel (j < 0) \parallel (k < 0)$
  for $i = 0; i \leq |x|$ **do**
    for $j = 0; j \leq |y|$ **do**
      for $k = 0; k \leq |z|$ **do**
        if $i = j = k = 0$ **then**
          $s(i,j,k) \leftarrow \{\epsilon\}$
        else
          $s(i,j,k) \leftarrow s(i-1,j-1,k)$ si $x(i) = y(j)$
          $\cup\, s(i-1,j,k-1)$ si $x(i) = z(k)$
          $\cup\, s(i,j-1,k).y(j)$
          $\cup\, s(i,j,k-1).z(k)$
        end
      end
    end
  end
end
return $s(|x|,|y|,|y|)$

Algorithme 5 : résout_équation_tab

Ces algorithmes tabulaires présentent la même complexité que les solveurs à base d'automates : en $O(|x| \times |y| \times |z| \times |t|)$ pour la vérification ; en $O(|x| \times |y| \times |z|)$ (en apparence) pour la résolution. Ici encore, si l'on veut pouvoir énumérer explicitement l'ensemble des solutions, la complexité est a priori exponentielle.

cf. Sec. B, p. 157

La procédure de recherche de triplets repose sur les mêmes types de constructions ; en particulier, le solveur à base de transducteurs doit construire $((L \bullet x) \backslash L) \cap L$ où x et L représentent respectivement le mot et le langage à partir desquels la recherche de triplets est effectuée. Les remarques concernant la complexité s'appliquent également.

4.1.4 Pondérations

Dans la section précédente, nous avons étudié deux solveurs d'équations analogiques, l'un fondé sur des transducteurs, et l'autre sur des méthodes de programmation dynamique. Les deux procédures associées sont non-déterministes par construction ; par conséquent, une équation peut conduire à plusieurs solutions. En outre, l'utilisation d'APPA implique la résolution de plusieurs équations dont les solutions sont agrégées ; savoir associer un poids à une proportion permet de comparer et d'ordonner des solutions concurrentes, de manière à, par exemple, n'en retenir qu'un sous-ensemble. Par ailleurs, les procédures impliquées dans la

résolution d'équation souffrent potentiellement d'une explosion combinatoire ; ne considérer que les meilleures solutions permet de les rendre plus efficaces. Toutes ces remarques soulignent l'importance de disposer de critères permettant de discriminer les proportions.

Nous exposons ici deux critères de « qualité » d'une proportion et quelques résultats associés.

Degré Le *degré* est une notion introduite initialement dans la définition 4. Elle correspond au nombre de facteurs alternant dans une proportion : plus le degré est petit, meilleure est la proportion. Les proportions incontestables sont les proportions atomiques, de la forme $a : a :: b : b$ ou $a : b :: a : b$, de degré 1. Puisque le degré correspond au nombre de découpages effectués dans les mots impliqués, un degré faible assure que les mots d'origine sont relativement préservés.

Le degré d'une proportion analogique $x : y :: z : t$ est noté $D(x : y :: z : t)$ et peut être caractérisé alternativement de la façon suivante.

Définition 8 (Degré). Le *degré* d'une proportion $x : y :: z : t$ est défini par :

$$D(x : y :: z : t) = \min_{(f_x, f_y, f_z, f_t) \in \mathcal{AF}} |f_x|.$$

Taille Le degré compte le nombre de facteurs alternant deux-à-deux. La *taille* d'une équation analogique compte, quant à elle, les alignements de symboles, qui prennent tous la forme $a : a :: b : b$ ou $a : b :: a : b$ (cf. figure 4.8).

Définition 9 (Taille). La *taille* d'une proportion analogique $x : y :: z : t$ est définie par :

$$T(x : y :: z : t) = \min_{(f_x, f_y, f_z, f_t) \in \mathcal{AF}} \sum_{i \in I_{f_x}} \max(|f_x(i)|, |f_t(i)|)$$

Pour illustrer ces deux notions, la figure 4.8 présente deux exemples. Alors que la proportion *chante* : *chantant* :: *parle* : *parlant* a un degré égal à 2 et une taille égale à 7, *chante* : *chantant* :: *parle* : *paantrle* a un degré de 4 et une taille de 11, en accord avec l'intuition que la première est meilleure que la seconde.

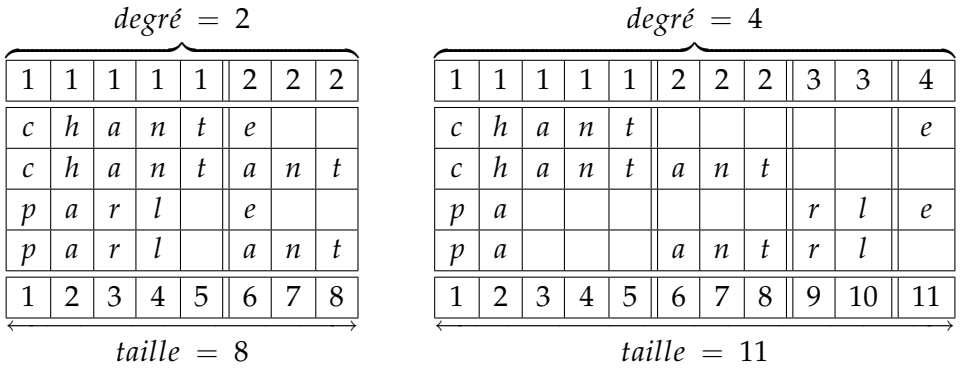


FIG. 4.8 – Le degré et la taille de deux proportions

ϵ-order Le ϵ-order d'une proportion correspond au nombre maximum de facteurs vides consécutifs apparaissant dans une factorisation. Dans l'exemple exposé plus haut, on observe que *chante* : *chantant* :: *parle* : *parlant* a un ϵ-order de 1 alors que *chante* : *chantant* :: *parle* : *paantrle* a un ϵ-order de 2, puisque deux facteurs vides consécutifs apparaissent dans les factorisations de *chante*, *chantant* et *parle* (cf. figure 4.8).

Définition 10 (ϵ-order). Le *ϵ-order* d'une proportion $x : y :: z : t$ relativement à $\alpha \in \{x, y, z, t\}$ est défini par :

$$O_\epsilon(x : y :: z : t,\ \alpha) = \min_{(f_x, f_y, f_z, f_t) \in \mathcal{AF}} \max_{J \subseteq I_{f_\alpha}} \{|J| : J \in \mathcal{I} \text{ et } f_\alpha(J) = \epsilon\}.$$

Pour la proportion entière, il devient :

$$O_\epsilon(x : y :: z : t) = \max_{\alpha \in \{x,y,z,t\}} \{O_\epsilon(x : y :: z : t,\ \alpha)\}$$

Une relation peut être établie entre ces trois notions, comme en témoigne le résultat suivant.

Proposition 7. Si t est une solution de l'équation $x : y :: z :?$ telle que

$$D(x : y :: z : t) > O_\epsilon(x : y :: z : t,\ x) > 1,$$

alors il existe une solution $\hat{t}$ telle que les (in)égalités suivantes sont vérifiées :

$$\begin{aligned} O_\epsilon(x : y :: z : \hat{t},\ x) &= O_\epsilon(x : y :: z : t,\ x) - 1, \\ D(x : y :: z : \hat{t}) &= D(x : y :: z : t) - 1, \\ T(x : y :: z : \hat{t}) &\leq T(x : y :: z : t). \end{aligned}$$

Démonstration. Soit une proportion analogique $x : y :: z : t$, avec

$$D = D(x : y :: z : t), O = O_\epsilon(x : y :: z : t,\ x) \text{ et } D > O > 1.$$

Par hypothèse, il existe des factorisations $(f_x, f_y, f_z, f_t) \in \mathcal{AF}(x : y :: z : t)$ de taille n et un intervalle $J = [\![i, j]\!]$ tels que $f_x(J) = \epsilon$, $|J| = O$ et pour tous les intervalles $I \subseteq [\![1, n]\!]$ avec $|I| > O$, $f_x(I) \neq \epsilon$. Puisque $D > O$, il existe un facteur non-vide dans x, donc on a ou bien $i > 1$, ou bien $j < D$. Pour des raisons de symétrie, on peut supposer $i > 1$ sans perte de généralité. La preuve implique la construction de nouvelles factorisations de x, y et z, obtenues par la permutation des facteurs à la position i et $i + 1$ dans x, y et z. Pour $\alpha \in \{x, y, z, t\}$, on pose

$$\begin{array}{lll} \hat{f}_\alpha(k) & = f_\alpha(k) & \text{si } k < i - 1, \\ \hat{f}_\alpha(i-1) & = f_\alpha(i-1) f_\alpha(i+1), & \\ \hat{f}_\alpha(i) & = f_\alpha(i), & \\ \hat{f}_\alpha(k) & = f_\alpha(k+1) & \text{si } i < k < D. \end{array}$$

Puisque $|J| > 1$, on a $i + 1 \in J$ et $f_x(i + 1) = \epsilon$. De plus, par symétrie, on peut supposer $(f_x(i), f_t(i)) = (f_y(i), f_z(i))$ et $(f_x(i+1), f_t(i+1)) = (f_z(i+1), f_y(i+1))$. En particulier, puisque $f_x(i) = f_y(i) = \epsilon$ et $f_x(i + 1) = f_z(i + 1) = \epsilon$, on peut

écrire $f_\alpha(i+1)f_\alpha(i) = f_\alpha(i-1)f_\alpha(i)$ pour $\alpha \in \{x, y, z\}$; en permutant ces facteurs, on obtient des factorisations de x, y et z. Le résultat de cette construction est une factorisation $\hat{f}_t$ pour un mot $\hat{t}$ telle que $x : y :: z : \hat{t}$ avec

$$D(x : y :: z : \hat{t}) = |\hat{f}_x| = |f_x| - 1 = D - 1.$$

De plus, l'ϵ-order « local » des nouvelles factorisations a diminué de 1, donc l'égalité $O_\epsilon(x : y :: z : \hat{t},\ x) = O_\epsilon(x : y :: z : t,\ x) - 1$ s'obtient en appliquant les mêmes permutations pour tous les intervalles J tels que $f_x(J) = \epsilon$ et $|J| = O$.

Enfin, puisque

$$(f_x(i), f_t(i)) = (f_y(i), f_z(i)) \text{ avec } f_x(i) = f_y(i) = \epsilon$$

et

$$(f_x(i+1), f_t(i+1)) = (f_z(i+1), f_y(i+1)) \text{ avec } f_x(i+1) = f_z(i+1) = \epsilon,$$

on a

$$\begin{aligned}
\max(|f_x(i-1)|, |f_t(i-1)|) + \max(|f_x(i)|, |f_t(i)|) & \\
+ \max(|f_x(i+1)|, |f_t(i+1)|) & \\
= \max(|f_x(i-1)|, |f_t(i-1)|) + |f_z(i)| + |f_y(i+1)| & \\
\leq \max(|f_x(i-1)|, |f_t(i-1)| + |f_y(i+1)|) + |f_z(i)|, &
\end{aligned}$$

ce qui donne

$$T(x : y :: z : \hat{t}) \leq T(x : y :: z : t). \qquad \square$$

Corollaire 1. *Si t est une solution de l'équation $x : y :: z :?$ telle que*

$$D(x : y :: z : t) > O_\epsilon(x : y :: z : t,\ x) > 1,$$

alors il existe une solution $\hat{t}$ telle que

$$O_\epsilon(x : y :: z : \hat{t},\ x) = 1.$$

Ce résultat implique, en particulier, qu'une recherche de solutions de degré minimal pour une équation $x : y :: z :?$ peut se limiter aux mots t qui vérifient $O_\epsilon(x : y :: z : t,\ x) \leq 1$; c'est ce qui permet en pratique d'éviter l'explosion combinatoire potentielle.

4.2 Une caractérisation algébrique des proportions analogiques

4.2.1 Introduction

Jusqu'ici, nous avons étudié le cas des proportions analogiques entre mots, éléments d'un monoïde libre. Dans cette section, nous montrons, dans un premier

temps, que la définition proposée s'étend naturellement au cas plus général des semigroupes et des magmas, fournissant un cadre pour une vaste gamme de structures algébriques, en particulier les groupes et les treillis. Ensuite, nous présentons un modèle de proportions analogiques « en cascade » permettant de traiter des situations plus riches, notamment le cas des proportions analogiques entre langages.

Rappels d'algèbre

Dans la suite de cette section, il sera question de structures algébriques. Nous rappelons ici les principales spécificités des structures dont il sera question. Par structure, nous entendons un ensemble muni d'une opération, dénommée loi de composition interne. Une structure est caractérisée par l'ensemble des propriétés que cette loi vérifie ; nous présentons dans la suite les propriétés les plus courantes.

Associativité Une loi $\oplus$ sur une structure S est *associative* ssi
- $\forall x, y, z \in S, (x \oplus y) \oplus z = x \oplus (y \oplus z)$.

Élément neutre Un *élément neutre* e (ou 1_S) sur une structure S est tel que :
- $\forall x, e \oplus x = x \oplus e = x$.

Commutativité Une loi $\oplus$ sur une structure S est *commutative* ssi
- $\forall x, y \in S, x \oplus y = y \oplus x$.

Liberté Une structure $(S, \oplus)$ possède la propriété de liberté ssi
- $\oplus$ est associative ;
- $\exists G \subseteq S$ t.q. $\forall x \in S, \exists!(x_1, x_2, \ldots, x_n) \in G$ t.q. $x = x_1 \oplus x_2 \oplus \cdots \oplus x_n$.

Inversion Une structure $(S, \oplus)$ possède la propriété d'inversion ssi
- $\forall x \in S, \exists -x$ t.q. $x \oplus -x = 1_S$.

La structure la plus générale est le *magma* : c'est un ensemble muni d'une opération interne, non nécessairement associative. Un *semigroupe* est un magma dont la loi de composition interne est associative. Un *monoïde* est un semigroupe avec un élément neutre et un monoïde libre est un monoïde possédant la propriété de liberté. La hiérarchie de quelques structures algébriques est représentée sur la figure 4.9.

4.2.2 Semigroupes, magmas et dérivés

cf. Sec. 4.1.2, p. 79

La définition proposée dans la section 4.1 pour traiter le cas des monoïdes libres se révèle directement applicable au cas des semigroupes sans modification. En effet, cette définition ne fait appel qu'à une loi de composition interne associative (la concaténation). Ainsi, la propriété de liberté des monoïdes libres, bien qu'elle autorise les constructions présentées dans la section 4.1.3, n'est pas nécessaire d'un point de vue uniquement définitionnel.

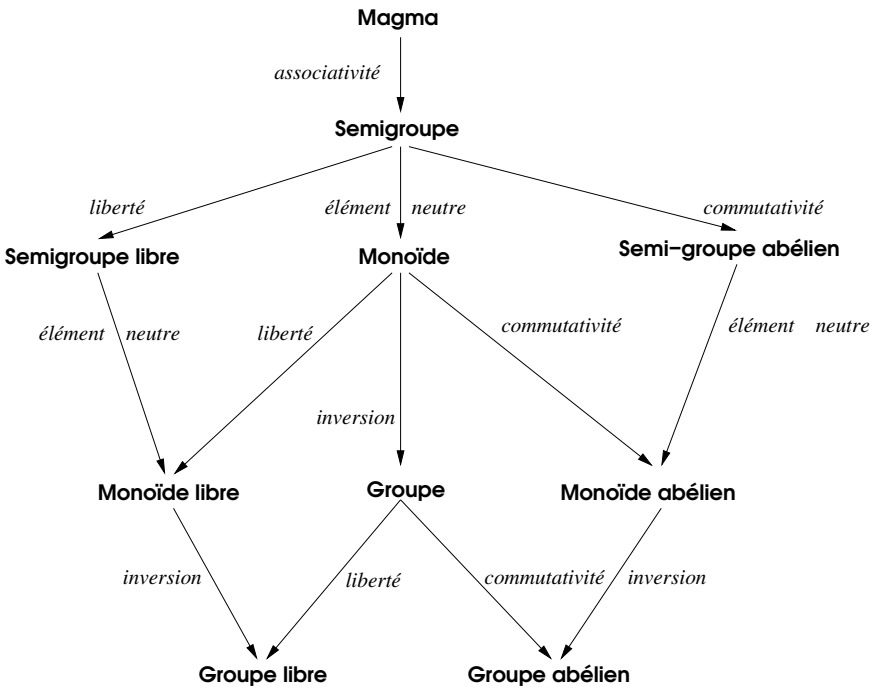


FIG. 4.9 – Hiérarchie de structures algébriques

En réalité, nous disposons donc d'un modèle de proportions analogiques applicable aux cas des semigroupes et a fortiori, à toutes les structures plus spécifiques. Dans la hiérarchie de la figure 4.9, nous sommes donc « remontés » du cas des monoïdes libres à celui des semigroupes, sans frais additionnel. Nous allons également voir que le modèle est également adaptable à peu de frais au cas encore plus général des magmas. À partir de cette position, nous pouvons alors « redescendre » dans l'intégralité du graphe, de manière à obtenir une définition applicable à l'ensemble de ces structures.

Semigroupes

La notion de factorisation, initialement formulée dans le cas des monoïdes libres, s'étend directement au cas des semigroupes.

Définition 11 (Factorisation). Une *factorisation* d'un élément u d'un semigroupe $(U, \oplus)$ est une séquence $(u_1, \ldots, u_m) \in U^m$ telle que $u_1 \oplus \ldots \oplus u_n = u$. Chaque terme u_i est un *facteur* de u.

Nous utiliserons les mêmes notations que celles introduites pour les factorisations de mots, à savoir $f_u(i) = u_i$ et $|f_u| = m$. Nous pouvons maintenant établir une définition de la proportion analogique entre éléments de semigroupes généralisant la définition 4.

Définition 12 (Proportion analogique (semigroupes)). Pour $(x, y, z, t) \in U^4$, on a $x : y :: z : t$ si et seulement s'il existe des factorisations $(f_x, f_y, f_z, f_t) \in (U^d)^4$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}.$$

Par construction, on a $|f_x| = |f_y| = |f_z| = |f_z| = d$. Le plus petit d pour lequel de telles factorisations peuvent être trouvées est nommé le *degré* de la proportion.

À coût nul, nous obtenons donc une définition applicable à tout semigroupe.

Généralisation au cas des magmas

Le passage du cas des monoïdes libres à celui des semigroupes s'est opéré sans difficulté. Nous voyons dans la suite comment traiter le cas encore plus général des magmas [8].

cf. Sec. 4.1.2, p. 79

Nous avons déjà souligné que la définition utilisée comme point de départ repose essentiellement sur deux mécanismes : la décomposition des objets en constituants plus petits, et une alternance deux-à-deux des constituants. La notion de factorisation, rendant compte de la décomposition dans le cas des monoïdes et des semigroupes, n'est pas applicable directement au cas des magmas : puisque l'opération interne d'une telle structure n'est pas associative, l'écriture $x_1 \oplus \cdots \oplus x_n$ n'est pas valide. La notion de factorisation doit par conséquent être adaptée de la façon suivante.

Définition 13 (Factorisation). Une *factorisation* d'un élément u d'un magma $(U, \oplus)$ est une expression générée par la grammaire hors-contexte décrite par les productions suivantes :

- (1): $\forall u \in U, S \to u$;
- (2): $S \to (\ S\ \oplus\ S\ )$.

Par exemple, $((2 + 3) + 5)$ est une factorisation de 10 sur le magma $(\mathbb{N}, +)$. Une telle factorisation est représentable par un arbre binaire (cf. figure 4.10). En effet, l'ensemble des factorisations sur U correspond en réalité à l'ensemble des arbres binaires dont les feuilles sont étiquetées par des éléments de U.

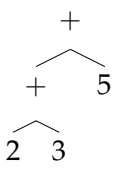


FIG. 4.10 – Une factorisation de 10 sur le magma $(\mathbb{N}, +)$

Le « squelette » de l'arbre est déterminé par l'application de la règle (2), et ses feuilles par l'application des règle de type (1). Nous dirons que deux factorisations sont *comparables* si elles ont le même squelette, c'est-à-dire si elles ont été générées

8. Notons que l'étude des magmas est en lien direct avec celle des arbres binaires.

par la même séquence de types de règles (toutes les séquences de type (1) sont vues de façon équivalente). Le nombre de feuilles de l'arbre est déterminé par le nombre d'applications de règles de type (1). Pour une factorisation f_x, nous noterons $f_x(i)$ l'étiquette de la feuille obtenue par la i^{e} application d'une telle règle. L'ensemble des factorisations sur U sera noté $\mathcal{F}(U)$. Cette notion est bien une généralisation de la notion de factorisation sur les semigroupes ; dans ce cas, en effet, les factorisations considérées sont des « peignes ». Les peignes comparables sont des peignes de même longueur. La *profondeur* de la factorisation correspond à la profondeur de l'arbre sous-jacent.

La définition de la proportion analogique est alors modifiée comme suit.

Définition 14 (Proportion analogique (cas des magmas)). Pour quatre éléments $(x, y, z, t) \in U^4$, on a $x : y :: z : t$ si et seulement s'il existe des factorisations comparables $(f_x, f_y, f_z, f_t) \in \mathcal{F}(U)^4$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}.$$

Proposition 8. La définition 14 peut être formulée récursivement. On a $x : y :: z : t$ si et seulement si

- ou bien $(y, z) \in \{(x, t), (t, x)\}$ (cas 1) ;
- ou bien $\exists(\alpha_1, \alpha_2) \in U^2$ pour $\alpha \in \{x, y, z, t\}$ tels que $\alpha = \alpha_1 \oplus \alpha_2$, et $\forall i \in [\![1, 2]\!]$, $x_i : y_i :: z_i : t_i$ (cas 2).

Démonstration. $\Leftarrow$. La définition récursive donnée implique la création de factorisations comparables de x, y, z, t, par applications des règles de type (1) (cas 1) et des règles de type (2) (cas 2). Ces factorisations vérifient bien les conditions d'alternance au niveau des feuilles ($(y, z) \in \{(x, t), (t, x)\}$).

$\Rightarrow$. Direct par récurrence sur la profondeur des factorisations comparables vérifiant les conditions d'alternance. □

Cette caractérisation nous permet de mettre en évidence une autre mesure d'une proportion analogique : la *profondeur*, égale à la profondeur des factorisations sous-jacentes à la proportion [9].

Limitations

D'un point de vue uniquement définitionnel, le modèle présenté précédemment apparaît satisfaisant car à la fois général et capable de couvrir les cas visés. En revanche, la présence de quantificateurs existentiels limite son application directe : il semble inenvisageable d'examiner exhaustivement l'ensemble des factorisations d'un élément dans le cas général, d'autant plus que celui-ci est potentiellement infini. Dans la suite, nous étudions un certain nombre de cas pour lesquels cette

9. Dans le cas des semigroupes, la forme des squelettes des factorisations n'est pas déterminante ; la profondeur minimale s'obtient alors en construisant un arbre dit quasi-complet. Dans ce cas particulier, la profondeur p est liée au degré d par la relation $p = \lceil ln_2(d) \rceil$.

situation se simplifie, conduisant à l'applicabilité du modèle. C'est le cas des monoïdes libres, pour lesquels la propriété de liberté a in fine permis la disparition des quantificateurs. Comme nous allons le voir, cette simplification est également possible pour d'autres structures, telles que les groupes abéliens et les treillis.

Semigroupes abéliens

Lorsque la loi de composition interne d'un semigroupe U est commutative, le semigroupe est dit commutatif ou *abélien*. Dans un semigroupe abélien, l'ordre des termes d'une factorisation d'un élément n'a pas d'importance. Dans le cas de la proportion, il est donc possible de réordonner les facteurs des factorisations d'une manière telle que les facteurs provenant de y dans x soient regroupés « à gauche », et les facteurs provenant de z dans x « à droite ». Le résultat suivant découle immédiatement de cette remarque.

Proposition 9 (Proportion analogique (dans un semigroupe abélien)). Pour un quadruplet $(x, y, z, t) \in U^4$, on a $x : y :: z : t$ ssi ou bien $(y, z) \in \{(x, t), (t, x)\}$, ou bien $\exists (x_1, x_2, t_1, t_2) \in U^4$ tels que $x = x_1 \oplus x_2,\ y = x_1 \oplus t_2, z = t_1 \oplus x_2,\ t = t_1 \oplus t_2$.

Démonstration. Par définition, il existe des factorisations $(f_x, f_y, f_z, f_t) \in (U^d)^4$ et un ensemble d'indices $I \subseteq [\![1, d]\!]$ tels que

$$f_x(I) = f_y(I), f_z(I) = f_t(I), f_x(J) = f_z(J), f_y(J) = f_t(J),$$

avec $J = [\![1, d]\!] \setminus I$.

Soit $\Lambda = \{x, y, z, t\}$. Par commutativité de l'opération $\oplus$, $\forall \alpha \in \Lambda$, $(f_\alpha(I), f_\alpha(J))$ est une factorisation de α, d'où le résultat recherché. □

Groupes abéliens

Un groupe abélien $(U, \oplus)$ est un monoïde abélien dans lequel tout élément x est associé à un unique élément $\ominus x$ tel que $x \oplus \ominus x = 1_U$, appelé son *inverse*.

Proposition 10 (Proportion analogique (cas des groupes abéliens)). Pour quatre éléments $(x, y, z, t) \in U^4$, on a $x : y :: z : t$ si et seulement si

$$x \oplus (\ominus y) = z \oplus (\ominus t).$$

Démonstration. La proposition 9 donne directement

$$x \oplus (\ominus y) = x_2 \oplus (\ominus t_2) = z \oplus (\ominus t).$$

Réciproquement, si $x \oplus (\ominus y) = z \oplus (\ominus t)$, en posant

$$x_1 = 1_U, x_2 = x, t_1 = z \oplus (\ominus x) \text{ et } t_2 = y,$$

on a

$$x = x_1 \oplus x_2, y = x_1 \oplus t_2, z = t_1 \oplus x_2 \text{ et } t = t_1 \oplus t_2.$$ □

Cette définition est cohérente avec l'idée traditionnelle et intuitive de la notion de proportion analogique. En particulier, sur l'ensemble des réels non nuls $(\mathbb{R}^\star, \times)$, elle correspond à la relation de proportionnalité $(\frac{x}{y} = \frac{z}{t})$; dans un espace vectoriel, elle exprime la relation entre les sommets d'un parallélogramme ($\vec{x} - \vec{y} = \vec{z} - \vec{t}$, cf. figure 4.11). Elle est donc en plein accord avec la proportion d'Euclide comme égalité de rapports (similarité de raisons).

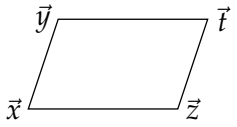


FIG. 4.11 – Un parallélogramme

Ici, il est important de noter que c'est la *même* définition de la proportion analogique qui nous permet de modéliser aussi bien la proportion entre les sommets d'un parallélogramme, la proportion

$$2 : 4 :: 7 : 9,$$

que celle qui suit :

$$\textit{indécorable} : \textit{décorer} :: \textit{indémontable} : \textit{démonter},$$

nous confortant quant à la cohérence du modèle.

4.2.3 Représentations structurées d'objets linguistiques

Les efforts de formalisation et d'abstraction effectués plus haut ne sont pas vains : le cadre algébrique abstrait présenté fournit un socle permettant d'offrir une définition de la proportion analogique convenant aux ensembles, structures de traits et langages, représentations couramment utilisées pour manipuler les données linguistiques.

Ensembles

L'opération d'union confère à l'ensemble des parties $(2^E, \cup)$ d'un ensemble E la structure de monoïde abélien : elle est associative, commutative, et l'ensemble vide $\emptyset$ est son élément neutre. En tant que monoïde abélien, la proposition 9 s'applique.

Proportion analogique (entre parties d'un ensemble) Pour $(x, y, z, t) \in (2^S)^4$, on a $x : y :: z : t$ si et seulement si $\exists (x_1, x_2, t_1, t_2) \in (2^S)^4$ tels que

$$x = x_1 \cup x_2,\ y = x_1 \cup t_2,\ z = t_1 \cup x_2,\ t = t_1 \cup t_2.$$

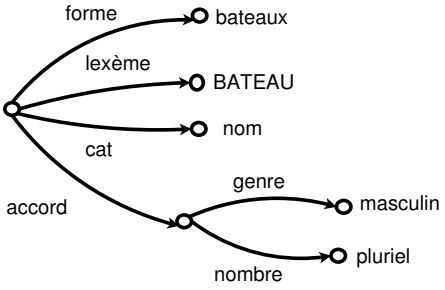
Structures de traits

Les *structures de traits* (ST) servent fréquemment de représentations aux données linguistiques. Une structure de traits est constituée d'un ensemble de traits auxquels on associe une valeur. Une valeur peut être *atomique*, ou *complexe* (i.e. c'est une ST elle-même). Prenons le cas des descripteurs linguistiques utilisés en morphologie flexionnelle (cf. section 3.4.2). Pour représenter une entrée d'un lexique flexionnel, on peut, par exemple, considérer les traits *forme*, *lexème*, *catégorie syntaxique*, *genre*, *nombre*, *temps*, etc., donnant lieu à la description (matricielle) suivante :

$$\begin{bmatrix} < forme >: bateaux \\ < lexème >: \texttt{BATEAU} \\ < cat >: \texttt{nom} \\ < accord >: \begin{bmatrix} < genre >: \texttt{masculin} \\ < nombre >: \texttt{pluriel} \end{bmatrix} \end{bmatrix}$$

Dans cet exemple, la valeur du trait *cat* est atomique, alors que celle du trait *accord* est complexe (la valeur du trait *accord* est elle-même une structure constituée des traits *genre* et *nombre*). Notons également que le trait *temps* n'est pas renseigné car non pertinent pour un nom. Les structures de traits sont également fortement impliquées dans différents modèles syntaxiques, tels que les LFG (*Lexical Functional Grammar*) ou les HPSG (*Head-Driven Phrase Structure Grammar*) ; Carpenter (1992) fournit une introduction complète aux ST et à leurs usages en linguistique computationnelle.

Les graphes acycliques orientés étiquetés fournissent un autre mode de représentation privilégié pour les ST [10]. Ainsi, la même structure est représentée par le graphe suivant :



Dans un tel graphe, un nœud peut avoir plusieurs arcs entrants (on parlera de *ré-entrance*), ce qui offre la possibilité à plusieurs traits de partager la même valeur.

Formellement, l'ensemble des ST, noté EST, est une algèbre booléenne, i.e. un ensemble muni de deux lois de composition interne, toutes deux associatives, idempotentes [11], commutatives et mutuellement distributives. Les lois de composition de l'ensemble EST sont l'*unification*, notée $\sqcup$ et la *généralisation*, notée $\sqcap$.

L'ensemble EST est muni d'une relation d'ordre partiel, la *subsomption*, notée $\sqsubseteq$. Si a et b sont deux ST, a subsume b si elle est plus générale qu'elle, c'est-à-dire

10. Carpenter (1992) autorise la présence de cycles dans les ST. Nous ne considérons pas ces cas dans ce travail, bien que la plupart des résultats présentés puissent également s'y appliquer.

11. $\forall x, x \sqcup x = x \sqcap x = x$.

si elle renseigne moins de traits que b tout en étant en accord avec elle sur ceux qu'elle renseigne. Par exemple, avec :

$$a = \begin{bmatrix} < forme >: bateaux \\ < cat >: \texttt{nom} \\ < accord >: \left[< genre >: \texttt{masculin} \right] \end{bmatrix} \text{ et } b = \begin{bmatrix} < forme >: bateaux \\ < lexème >: \texttt{BATEAU} \\ < cat >: \texttt{nom} \\ < accord >: \begin{bmatrix} < genre >: \texttt{masculin} \\ < nombre >: \texttt{pluriel} \end{bmatrix} \end{bmatrix}$$

on a $a \sqsubseteq b$.

L'opération d'unification entre deux structures a et b consiste à trouver la structure la plus générale (au sens de la subsomption) qui est plus spécifique à la fois que a et b. Intuitivement, unifier deux structures de traits consiste à fusionner leurs informations ; si ces informations sont contradictoires, le résultat de l'unification est un élément $\top$ (top), ajouté de façon à ce que l'opération d'unification soit toujours définie. La structure vide $\bot$ (bottom) est l'élément neutre pour l'unification et l'élément top $\top$ est l'élément neutre pour la généralisation. Puisqu'une algèbre booléenne est clairement un cas particulier de semigroupe abélien (pour les deux opérations), la proposition 9 s'applique de nouveau, fournissant une définition de la proportion analogique entre structures de traits.

Proportions analogiques (entre structures de traits) Pour $(x, y, z, t) \in ST^4$, on a $x : y :: z : t$ si et seulement si $\exists (x_1, x_2, t_1, t_2) \in ST^4$ tels que

$$x = x_1 \sqcup x_2,\ y = x_1 \sqcup t_2,\ z = t_1 \sqcup x_2,\ t = t_1 \sqcup t_2.$$

Remarquons que cette définition est valide quel que soit le formalisme associé aux structures de traits impliquées ; en particulier, elles peuvent être sujettes à la réentrance. Dans cette définition, les seules propriétés exploitées sont l'associativité et la commutativité de l'opération d'unification. En outre, signalons que les différents exemples exposés dans la section 4 (p. 75) sont correctement modélisés.

Treillis

Le problème que pose la définition générique donnée à la section 4.2.2 réside dans l'introduction d'une quantification existentielle. Nous avons pu faire disparaître cette dernière dans les cas particuliers des monoïdes libres et des groupes abéliens. Nous présentons dans la suite une simplification similaire lorsque la structure algébrique considérée est un treillis. Les deux exemples de représentations linguistiques présentés ci-dessus, à savoir les ensembles et les structures de traits, forment des treillis et bénéficieront donc des simplifications que nous allons introduire.

Un *treillis* est une structure mathématique qui peut être vue soit comme un ensemble partiellement ordonné (*partially ordered set* en anglais, poset dans la suite), soit comme une algèbre. Un treillis $(L; \leq)$ peut être défini comme un poset dans lequel il est possible d'associer à tout couple d'élément un *supremum* et un *infimum*. Le supremum $s = \sup\{a, b\}$ de (a, b) est tel que

– $s \leq a$ et $s \leq b$;
– $\forall s' \in L, s' \leq a$ et $s' \leq b \Rightarrow s' \leq s$.
L'infimum se définit par dualité.

Par ailleurs, une algèbre $(L; \wedge, \vee)$ est un treillis si L est un ensemble non-vide, $\wedge$ et $\vee$ deux opérations binaires, toutes deux idempotentes, commutatives, associatives et vérifiant la loi d'absorption [12]. Tout treillis défini comme une algèbre peut être réexprimé en tant que treillis défini comme un poset, et vice-versa.

Théorème 1. *1. Soit $\mathcal{L} = (L; \leq)$ un poset vérifiant les propriétés des treillis. Soit $\wedge$ et $\vee$ les opérations définies par :*
– $\forall a, b \in L, a \wedge b = \inf\{a, b\}$;
– $\forall a, b \in L, a \vee b = \sup\{a, b\}$.
Alors, $(L; \wedge, \vee) = \mathcal{L}^a$ définit une algèbre, et cet algèbre est un treillis.

2. *Soit $\mathcal{L} = (L; \wedge, \vee)$ une algèbre vérifiant les propriétés des treillis. Soit $\leq$ la relation binaire définie par $a \leq b$ ssi $a \wedge b = a$ (ou de manière équivalente, $a \vee b = b$). Alors, $(L; \leq) = \mathcal{L}^p$ est un poset, et ce poset est un treillis.*
3. *Soit le treillis $\mathcal{L} = (L; \leq)$ vu comme un poset ; alors, $(\mathcal{L}^p)^a = \mathcal{L}$.*
4. *Soit le treillis $\mathcal{L} = (L; \wedge, \vee)$ vu comme une algèbre ; alors, $(\mathcal{L}^a)^p = \mathcal{L}$.*

L'ensemble des structures de traits est un treillis vu comme une algèbre relativement aux opérations d'unification et de généralisation. C'est un treillis vu comme un poset relativement à l'opération de subsomption. Pour l'ensemble des parties, ce sont respectivement les opérations d'union, d'intersection et d'inclusion qui jouent des rôles analogues.

Dans la suite, nous exploitons les propriétés particulières des treillis de manière à simplifier la définition de la proportion analogique, simplification qui s'appliquera en particulier aux cas des ensembles et des structures de traits.

Proposition 11 (Proportion analogique (cas des treillis)). Pour quatre éléments $(x, y, z, t) \in (L; \wedge, \vee)^4$, on a $x : y :: z : t$ si et seulement si :

$$\begin{aligned} x &= (x \wedge y) \vee (x \wedge z), \\ y &= (x \wedge y) \vee (t \wedge y), \\ z &= (t \wedge z) \vee (x \wedge z), \\ t &= (t \wedge z) \vee (t \wedge y). \end{aligned}$$

Démonstration. $\Leftarrow$. Trivial puisque la formulation de la proposition 11 est clairement un cas particulier de la proposition 9.

$\Rightarrow$. Soit des éléments $x, y, z, t \in L$ tels que $x : y :: z : t$. Par définition, il existe des éléments $x_1, x_2, t_1, t_2 \in L$ tels que

$$x = x_1 \vee x_2, y = x_1 \vee t_2, z = t_1 \vee x_2 \text{ et } t = t_1 \vee t_2.$$

Nous allons montrer $x = (x \wedge y) \vee (x \wedge z)$. Les inégalités

$$(x \wedge y) \vee (x \wedge z) \leq x \wedge (y \vee z) \leq x$$

12. $\forall a, b : a \wedge (a \vee b) = a \vee (a \wedge b) = a$.

s'obtiennent par la simple application des propriétés fondamentales du supremum et de l'infimum. Si $a \leq b$ et $c \leq d$, alors $a \leq b \leq (b \vee d)$ et $c \leq d \leq (b \vee d)$, ce qui donne $a \vee c \leq b \vee d$. Puisque $x_1 \leq x$ et $x_1 \leq y$, on a $x_1 \leq (x \wedge y)$. Symétriquement, on a $x_2 \leq (x \wedge z)$. Par conséquent,

$$x = x_1 \vee x_2 \leq (x \wedge y) \vee (x \wedge z).$$

Nous obtenons finalement

$$x = (x \wedge y) \vee (x \wedge z)$$

et les autres égalités sont vérifiées par symétrie. □

Dans le cas des treillis, la vérification de proportion n'implique donc que 8 opérations élémentaires.

Produits cartésiens

Un élément d'un produit cartésien d'ensembles quelconques (un vecteur non-homogène dans un contexte informatique) peut être représenté par une structure de traits plate (non récursive), dans laquelle tous les traits sont renseignés. Dans ce cas, la notion de proportion se simplifie.

Proposition 12 (Proportion analogique (cas des produits cartésiens)). Pour un quadruplet $(x, y, z, t) \in (E_1 \times \cdots \times E_n)^4$, on a $x : y :: z : t$ si et seulement si :

$$\forall i \in [\![1, n]\!], (y[i], z[i]) \in \{(x[i], t[i]), (t[i], x[i])\},$$

où $\alpha[i]$ représente la valeur de α sur le i^{e} trait, c'est-à-dire sa projection sur E_i.

Démonstration. $\Rightarrow$. Si $x = (x \wedge y) \vee (x \wedge z)$, alors

$$\forall i \in [\![1, n]\!], x[i] = ((x \wedge y) \vee (x \wedge z))[i] = (x[i] \wedge y[i]) \vee (x[i] \wedge z[i]).$$

Or, $a \wedge b = a$ si $a = b$, et $\bot$ sinon. Puisque $x[i]$ est égal à $(x[i] \wedge y[i]) \vee (x[i] \wedge z[i])$, l'un des deux termes $x[i] \wedge y[i]$ ou $x[i] \wedge z[i]$ n'est pas égal à $\bot$, ce qui implique $x[i] = y[i]$ ou $x[i] = z[i]$. Le reste du résultat s'obtient par symétrie.

$\Leftarrow$. On a $\forall i \in [\![1, n]\!], x[i] : y[i] :: z[i] : t[i]$. La proposition s'obtient directement en remarquant que pour $\alpha \in \{x, y, z, t\}$, on a $\alpha = \vee_i \alpha[i]$. □

Dans ce cas, la vérification et la résolution d'équation s'opèrent directement par examen de chacun des traits. En ce qui concerne la recherche de triplets, la procédure introduite pour les chaînes de symboles peut être exploitée. En effet, un vecteur (symbolique) de taille fixe peut être vu comme un mot sur l'alphabet composé des valeurs que prennent les attributs ; voir l'annexe B pour davantage de détails.

La proportion entre structures de traits et éléments d'un produit cartésien formalise plus clairement la notion d'analogie opérant parallèlement sur plusieurs

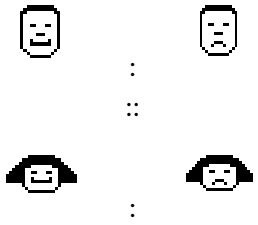
niveaux ; en particulier, elle permet de mieux comprendre l'utilisation des projections introduites dans le chapitre 3, à savoir

cf. Sec. 3.3.2, p. 58

$$x : y :: z : t \Rightarrow i(x) : i(y) :: i(z) : i(t) \text{ et } o(x) : o(y) :: o(z) : o(t).$$

Voir également Yvon (1999) pour une discussion sur de telles proportions multi-niveaux.

À titre illustratif, voici une analogie entre images, vues comme des vecteurs de pixels, identifiable à partir de la définition donnée ci-dessus :



4.2.4 Proportions en « cascade »

Dans le cadre algébrique présenté ci-dessus, la notion de proportion repose essentiellement sur : (i) la décomposition des termes en facteurs, (ii) l'alignement de ces facteurs en schémas spécifiques. En effet, dans ce cadre, quatre objets (x, y, z, t) forment une proportion analogique si et seulement si il existe des factorisations (f_x, f_y, f_z, f_t) de (x, y, z, t) telles que :

$$\forall i \in [\![1, d]\!],\ (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}.$$

Cette dernière condition impose en particulier que les alignements entre les facteurs prennent l'une des deux formes $a : a :: b : b$ ou $a : b :: a : b$, c'est-à-dire la forme d'une proportion atomique. Une façon de voir le modèle consiste à considérer que l'on forme des proportions (complexes) sur des objets structurés à partir de proportions de base plus simples (les proportions atomiques [13]). En considérant des proportions de base plus riches, ainsi qu'en autorisant l'application récursive de ce schéma de composition, il est possible d'étendre le modèle et de l'appliquer à des situations plus complexes. Ainsi, si l'on ne fait pas d'hypothèse supplémentaire sur les proportions de base, on peut généraliser la définition 12 de la façon suivante.

Définition 15 (Proportion analogique (en cascade)). Pour $(x, y, z, t) \in U^4$, on a $x : y :: z : t$ si et seulement s'il existe des factorisations $(f_x, f_y, f_z, f_t) \in (\mathcal{F}(U)))^4$ telles que

$$\forall i \in [\![1, d]\!], (f_x(i), f_y(i), f_z(i), f_z(i)) \in \mathcal{B}(U),$$

où $\mathcal{B}(U)$ représente l'ensemble des *proportions de base*.

13. L'utilisation de l'adjectif atomique est donc justifiée ; atomique s'oppose ici à complexe.

En notant $\mathcal{A}(U)$ l'ensemble des proportions atomiques, la définition 12 est clairement un cas particulier de la définition 15 pour lequel $\mathcal{A}(U) = \mathcal{B}(U)$. En revanche, $\mathcal{B}(U)$ peut maintenant contenir davantage que les seules proportions atomiques.

Pour illustrer ce mécanisme, prenons l'exemple de la proportion analogique entre séquences de phonèmes suivante :

/bɔ̃/ : /bɔnaʒ/ :: /mwajɛ̃/ : /mwajɛnaʒ/ (*bon* : *bon âge* :: *moyen* : *moyen-âge*)

En conservant la définition proposée dans la section 4.1, il n'est pas possible de modéliser cette proportion car l'alignement $(/ɔ̃/,/ɔ/,/ɛ̃/,/ɛ/)$ ne forme pas une proportion atomique. En revanche, si l'on considère la structure de l'ensemble des phonèmes, induite par une représentation sous forme de vecteurs de traits distinctifs, il semble légitime de vouloir tenir compte d'une proportion de la sorte, qui exprime simplement une opposition nasal/oral (opposition observable par ailleurs dans la graphie de l'alphabet phonétique international (IPA)).

Formellement, on est conduit à considérer une notion de proportion analogique entre les éléments d'un alphabet Σ (les phonèmes) couvrant davantage que les proportions atomiques, impliquant une notion enrichie de la proportion analogique entre éléments de $\Sigma^\star$ (les séquences de phonèmes).

La caractérisation $\mathcal{B}(U)$ peut provenir d'une connaissance spécifique ; dans la suite, nous étudions le cas où une définition de la proportion analogique est disponible sur une partie $P \subseteq U$. Une proportion entre quatre objets $(x, y, z, t) \in P^4$ relativement à P est notée $x : y \overset{P}{::} z : t$; dans ce cas, $\mathcal{B}(U)$ est alors simplement défini par

$$\mathcal{B}(U) = \{(x, y, z, t) \in P^4 |\ x : y \overset{P}{::} z : t\}.$$

Pour modéliser l'exemple précédent, il s'agit d'exploiter les proportions analogiques sur $\Sigma \subseteq \Sigma^\star$. Plus précisément, les étapes opérées sont les suivantes.

– Les phonèmes sont représentés par des vecteurs de traits distinctifs appartenant à l'ensemble (fini)

$$F = \{occlusif, sonore, voisé, nasal, \ldots\};$$

l'ensemble des phonèmes est $\Sigma \subset ST(F)$. La définition de la proportion analogique entre structures de traits (cf. section 4.2.3) peut être utilisée pour caractériser l'analogie entre phonèmes.
– Cette définition (sur $\Sigma \subset \Sigma^\star$) sert de support pour construire l'ensemble des proportions de base sur $\Sigma^\star$.
– La définition de proportion en cascade s'applique finalement, de façon à obtenir un modèle de proportions sur $\Sigma^\star$.

Puisque nous disposons de définitions de proportions entre séquences, ensembles, structures de traits, ce schéma nous permet de définir des proportions entre séquences d'ensembles (comme ci-dessus), séquences de structures de traits,

ensembles de séquences, etc. Le cas des séquences de symboles quelconques est traité en détail dans Yvon *et al.* (2004) ; en particulier, la question suivante y est traitée : comment des coûts associés à des proportions de base (entre symboles) permettent-ils de définir des coûts sur des proportions plus complexes (entre séquences) ? Le cas des langages, ensembles de séquences, est étudié ci-dessous.

Langages

Les langages (finis) sont des représentations d'entités linguistiques fréquemment utilisées dans les applications de TAL. Ils fournissent notamment un modèle naturel pour représenter les ambiguïtés. Par exemple, ils peuvent représenter les analyses possibles d'une forme graphique, ou les différentes façons de la prononcer. Un exemple de telle proportion est illustré sur la figure 4.12.

(potato) [pəˈtɑːtə] : [pəˈtɑːtə], [pəˈteɪtə]
::
(tomato) [təˈmɑːtə] : [təˈmɑːtə], [təˈmeɪtə]

FIG. 4.12 – Proportion analogique entre langages

Un langage (fini) sur un alphabet Σ est un sous-ensemble (fini) de $\Sigma^\star$. L'ensemble des langages est un monoïde abélien relativement à l'opération classique d'union. En outre, c'est également un monoïde relativement à l'opération de concaténation, définie par :

$$L.K = \{lk \mid l \in L, k \in K\}.$$

En outre, la concaténation est distributive (à gauche et à droite) par rapport à l'union, i.e.

$$I.(J \cup K) = (I.J) \cup (I.K) \text{ et } (I \cup J).K = (I.K) \cup (J.K),$$

et l'élément neutre ($\emptyset = 0_K$) pour l'union est un zéro pour la concaténation :

$$\forall k \in K, k.0_K = 0_K.k = 0_K.$$

L'ensemble des langages vérifie donc les propriétés d'un semi-anneau.

Cela nous permet de voir les langages de deux façons différentes : (i) comme des ensembles de mots ; (ii) comme des éléments d'un semi-anneau. À partir de ces deux visions, il est possible de proposer deux définitions de la proportion analogique entre langages. En effet, pour la première, on peut appliquer le schéma en cascade étudié ci-dessus ; pour la seconde, on peut adapter la définition 8 au cas des structures à deux opérations. Dans la suite, nous étudions ces deux définitions plus en détail, et montrons qu'elles sont en réalité équivalentes, ce qui conforte une nouvelle fois la cohérence du modèle.

Les langages comme ensembles de mots

L'ensemble des langages sur un alphabet Σ est $2^{\Sigma^\star}$. Un langage étant un ensemble de mots, le schéma en cascade présenté plus haut est directement appli-

cable : $\mathcal{B}(2^{\Sigma^\star})$ est construit à partir de la définition de l'analogie entre mots, et par extension, entre singletons de $2^{\Sigma^\star}$. De façon à distinguer clairement les proportions entre mots et celles entre langages, les premières seront notées $x : y \overset{\Sigma^\star}{::} z : t$. Par exemple, puisque *searching* : *researcher* $\overset{\Sigma^\star}{::}$ *viewing* : *reviewer*, on a également $\{searching\} : \{researcher\} :: \{viewing\} : \{reviewer\}$ (sur $2^{\Sigma^\star}$). Cela s'exprime formellement par [14] :

$$\mathcal{B}(2^{\Sigma^\star}) = \{(\{x\},\{y\},\{z\},\{t\}) \mid (x,y,z,t) \in (\Sigma^\star)^4,\ x : y \overset{\Sigma^\star}{::} z : t\} \cup \mathcal{A}(2^{\Sigma^\star})$$

Nous nous sommes ici servi de la proportion entre mots comme tremplin au cas des langages. On peut aisément vérifier que

$$\begin{aligned}
\{\text{pəˈtɑːtə}\} &= \{\text{pəˈtɑːtə}\} \cup \{\text{pəˈtɑːtə}\} \\
\{\text{pəˈtɑːtə, pəˈteɪtə}\} &= \{\text{pəˈtɑːtə}\} \cup \{\text{pəˈteɪtə}\} \\
\{\text{təˈmɑːtə}\} &= \{\text{təˈmɑːtə}\} \cup \{\text{təˈmɑːtə}\} \\
\{\text{təˈmɑːtə, təˈmeɪtə}\} &= \{\text{təˈmɑːtə}\} \cup \{\text{təˈmeɪtə}\}
\end{aligned}$$

avec

$$\text{pəˈtɑːtə} : \text{pəˈtɑːtə} \overset{\Sigma^\star}{::} \text{təˈmɑːtə} : \text{təˈmɑːtə}$$

et

$$\text{pəˈtɑːtə} : \text{pəˈteɪtə} \overset{\Sigma^\star}{::} \text{təˈmɑːtə} : \text{təˈmeɪtə}$$

La dernière proportion est donc utilisée comme une proportion de base, mais elle est plus complexe qu'une proportion atomique. Cette définition de la proportion analogique entre langages sera désignée par le terme *WLang* et notée $x : y \overset{W}{::} z : t$.

Les langages comme éléments d'un semi-anneau

La notion de factorisation introduite pour traiter le cas des magmas et des semigroupes ne prend en compte qu'une seule opération ; elle est naturellement modifiable au cas des semi-anneaux [15].

Définition 16 (Factorisation). Une *factorisation* d'un élément u d'un semi-anneau $(U, +, \times)$ est une expression générée par la grammaire hors-contexte décrite par les règles de réécriture suivantes :

- (1): $\forall u \in U, S \to u$;
- (2): $S \to (\ S\ +\ S\ )$.

14. Puisque les proportions atomiques ont été aux fondements du cadre, il semble raisonnable d'imposer $\mathcal{A}(U) \subseteq \mathcal{B}(U)$; c'est la raison pour laquelle $\mathcal{A}(2^{\Sigma^\star})$ est explicitement ajouté à $\mathcal{B}(2^{\Sigma^\star})$.

15. La notion de factorisation entre langages porte usuellement un autre sens (voir par exemple Sakarovitch (2003)) ; nous ne changerons toutefois pas de vocabulaire, de façon à conserver une cohérence au sein du chapitre.

– (3): $S \rightarrow (\ S\ \times\ S\ )$.

Par exemple, $((\{a,b\} \cup \{c\}).\{a,c\})$ « factorise » le langage $\{aa, ba, ca, ac, bc, cc\}$ (cf. figure 4.13). La notion de proportion s'en déduit immédiatement. On obtient de

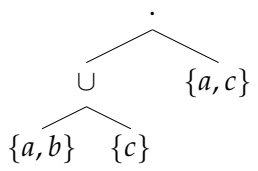


FIG. 4.13 – Une factorisation de $\{aa, ba, ca, ac, bc, cc\}$ sur le semi-anneau $(\Sigma^\star, \cup, .)$

cette façon une extension assez naturelle du modèle proposé initialement : puisque la structure sur laquelle on travaille est munie de deux opérations, ces deux opérations peuvent être utilisées pour « agréger » les « composants ».

Pour vérifier que l'exemple de proportion entre langages précédemment évoqué est également couvert dans ce modèle, il suffit d'observer :

$$\begin{aligned}
\{\ \text{pəˈtɑːtə}\} &= \{\ \text{pəˈt}\}\ .\ \ \{\ \text{ɑː}\}\ \ .\ \{\ \text{tə}\}\\
\{\ \text{pəˈtɑːtə, pəˈteItə}\} &= \{\ \text{pəˈt}\}\ .\ \{\ \text{ɑː, eI}\}\ .\ \{\ \text{tə}\}\\
\{\ \text{təˈmɑːtə}\} &= \{\ \text{təˈm}\}\ .\ \ \{\ \text{ɑː}\}\ \ .\ \{\ \text{tə}\}\\
\{\ \text{təˈmɑːtə, təˈmeItə}\} &= \{\ \text{təˈm}\}\ .\ \{\ \text{ɑː, eI}\}\ .\ \{\ \text{tə}\}.
\end{aligned}$$

Cette définition de la proportion analogique entre langages sera désignée par le terme *SLang* et notée $x : y \overset{S}{::} z : t$.

Résultat d'équivalence

Avant de présenter le résultat d'équivalence, comparons les deux définitions de manière informelle. Dans les deux cas, deux mécanismes sont utilisés : une définition des proportions de base et un mécanisme de composition. Dans le cas de *WLang*, l'ensemble des proportions de base contient les proportions atomiques ainsi que les proportions entre singletons et la composition est effectuée à l'aide de l'opération d'union. À l'opposé, dans le cas de *SLang*, les proportions de base sont les seules proportions atomiques ; en revanche, les deux opérations d'union et de concaténation sont utilisables dans le mécanisme de composition. Les proportions de base sont plus riches dans *WLang* et le mécanisme de composition plus riche dans *SLang*.

Proposition 13. $\forall (x,y,z,t) \in (2^{\Sigma^\star})^4, x : y \overset{W}{::} z : t \Leftrightarrow x : y \overset{S}{::} z : t.$

Démonstration. $\Rightarrow$. Par récurrence sur le degré de la proportion. Soit quatre termes $(x,y,z,t) \in (2^{\Sigma^\star})^4$ tels que $x : y \overset{W}{::} z : t$. Par définition, il existe des langages

$$(x_i)_{i\in[\![1,d]\!]}, (y_i)_{i\in[\![1,d]\!]}, (z_i)_{i\in[\![1,d]\!]}, (t_i)_{i\in[\![1,d]\!]}$$

t.q. $\forall \alpha \in \{x, y, z, t\}$, $\alpha_1 \cup \cdots \cup \alpha_d = \alpha$, et $\forall i \in [\![1, d]\!]$, $x_i : y_i :: z_i : t_i$ (proportion atomique ou proportion entre singletons). Si $d = 1$, alors le quadruplet (x, y, z, t) forme ou bien une proportion atomique ou bien une proportion entre singletons. Dans le premier cas, le résultat est trivialement vérifié. Si x, y, z et t sont des singletons, on peut noter $(x, y, z, t) = (\{x'\}, \{y'\}, \{z'\}, \{t'\})$ avec $x' : y' \overset{\Sigma^\star}{::} z' : t'$. La définition de la proportion analogique entre mots indique que pour $\alpha' \in \{x', y', z', t'\}$, il existe des facteurs $(\alpha'_i)_{i \in [\![1,d]\!]} \in \Sigma^\star$ tels que $\alpha' = \alpha'_1 \ldots \alpha'_m$ et

cf. Sec. 4.1.2, p. 79

$$\forall i, (y'_i, z'_i) \in \{(x'_i, t'_i), (t'_i, x'_i)\}.$$

En posant $\alpha_i = \{\alpha'_i\}$, nous avons aussi : $\alpha = \alpha_1 \ldots \alpha_m$ (par concaténation des langages) avec $\forall i$, $(y_i, z_i) \in \{(x_i, t_i), (t_i, x_i)\}$ et $\alpha_i \neq \emptyset$. Puisque la définition de *SLang* autorise l'utilisation de la concaténation pour l'agrégation, on a bien $x : y :: z : t$, ce qui montre l'implication pour $d = 1$. Supposons maintenant que l'implication est vérifiée pour tout degré $k < d$. Dans ce cas, l'implication est directe puisque le mécanisme de composition est plus riche dans le cas de *SLang*, ce qui prouve le résultat par récurrence.

$\Leftarrow$. Par récurrence sur la profondeur des factorisations sous-jacentes à la proportion. Soit $(x, y, z, t) \in (2^{\Sigma^\star})^4$ tels que $x : y \overset{S}{::} z : t$; p désigne la profondeur des factorisations sous-jacentes à la proportion. Si $p = 1$, alors le résultat est trivial puisque $A(2^{\Sigma^\star}) \subseteq B(2^{\Sigma^\star})$. Supposons maintenant que l'implication est vérifiée pour toute profondeur $k < p$. Par définition, $\exists x_1, x_2, t_1, t_2$ tels que $x = x_1 \oplus x_2, y = x_1 \oplus t_2, z = t_1 \oplus x_2, t = t_1 \oplus t_2$ avec $\oplus \in \{\cup, .\}$, $x_1 : y_1 \overset{S}{::} z_1 : t_1$ et $x_2 : y_2 \overset{S}{::} z_2 : t_2$. Si $\oplus = \cup$, le résultat est direct puisque l'union est autorisée dans le mécanisme d'agrégation de *WLang*. Si $\oplus = .$, alors, par hypothèse de récurrence, il est possible d'exprimer les proportions $x_1 : y_1 \overset{S}{::} z_1 : t_1$ et $x_2 : y_2 \overset{S}{::} z_2 : t_2$ comme des unions de proportions de base relativement à *WLang*, i.e. des proportions atomiques ou des proportions entre singletons. Par ailleurs, une proportion atomique entre des ensembles non-vides peut être réexprimée comme une union de proportions entre singletons. En effet, si $A : A :: B : B$ avec $A = \cup_{k=1}^{i} \{a_k\}$ et $B = \cup_{k=1}^{j} \{b_k\}$, alors on peut poser $m = max(i, j)$, $a_k = a_1$ pour $k \in [\![i + 1, m]\!]$ et $b_k = b_1$ pour $k \in [\![j + 1, m]\!]$, conduisant à $\forall k \in [\![1, m]\!]$, $\{a_i\} : \{a_i\} :: \{b_i\} : \{b_i\}$. Les proportions $x_1 : y_1 \overset{S}{::} z_1 : t_1$ et $x_2 : y_2 \overset{S}{::} z_2 : t_2$ peuvent dont être exprimées comme des unions de proportions entre singletons, i.e.

$$\forall \alpha_1 \in \{x_1, y_1, z_1, t_1\}, \alpha_1 = \cup_{k=1}^{m_1} \{\alpha'_{1k}\},\ \forall \alpha_2 \in \{x_2, y_2, z_2, t_2\}, \alpha_2 = \cup_{l=1}^{m_2} \{\alpha'_{1k}\},$$

avec $\forall k \in [\![1, m_1]\!]$, $x'_{1k} : y'_{1k} \overset{\Sigma^\star}{::} z'_{1k} : t'_{1k}$ et $\forall l \in [\![1, m_2]\!]$, $x'_{2k} : y'_{2k} \overset{\Sigma^\star}{::} z'_{2k} : t'_{2k}$. On a

$$\alpha = \alpha_1 \alpha_2 = \cup_{k=1,l=1}^{m_1,m_2} \{\alpha'_{1k}\}\{\alpha'_{2l}\} = \cup_{k=m_1,l=m_2}^{i,j} \{\alpha'_{1k}\alpha'_{2l}\}.$$

Puisque $\forall k \in [\![1, m_1]\!]$, $x'_{1k} : y'_{1k} \overset{\Sigma^\star}{::} z'_{1k} : t'_{1k}$ et $\forall l \in [\![1, m_2]\!]$, $x'_{2k} : y'_{2k} \overset{\Sigma^\star}{::} z'_{2k} : t'_{2k}$, on a $\forall (k, l) \in [\![1, m_1]\!] \times [\![1, m_2]\!]$, $x'_{1k}x'_{2l} : y'_{1k}y'_{2l} \overset{\Sigma^\star}{::} z'_{1k}z'_{2l} : t'_{1k}$. La proportion initiale peut donc être réexprimée comme une union de singletons formant des proportions, ce qui termine la preuve. □

Ce résultat d'équivalence souligne une nouvelle fois la cohérence du cadre adopté. En revanche, la définition ne fournit pas directement un modèle calculatoire exploitable, la présence de quantificateurs amenant potentiellement à une explosion combinatoire.

De la structure à l'algorithme

Les proportions analogiques, telles qu'étudiées dans cette section, reposent principalement sur les notions de décomposition et d'alternance. Ce cadre algébrique est bien adapté au cas « simple » des semigroupes, pour lesquels on ne considère qu'un unique opérateur. Deux directions ont déjà été explorées de manière à rendre ce modèle plus expressif : (i) l'utilisation de cascades, qui empilent récursivement des proportions ; (ii) l'ajout d'opérateurs, qui enrichissent les modes de composition. En particulier, de telles extensions ont été introduites pour traiter le cas des langages (cf. section 4.2.4) ; elles seront également nécessaires pour traiter celui des arbres (cf. section 4.2.5).

Si aucune restriction n'est opérée sur la nature des opérateurs, et si un nombre illimité de récursions est autorisé, le modèle devient expressif au point de pouvoir traiter n'importe quel objet issu de l'application d'une fonction calculable. Dans ce contexte, une factorisation d'un objet x n'est rien d'autre qu'un moyen algorithmique pouvant produire x. Ainsi, $u(x_1, \ldots, x_n) = x$ est une « factorisation algorithmique » de x ; le « squelette » de la factorisation est un algorithme (u), et les « feuilles » ses arguments $(x_1, \ldots, x_n)$. Des factorisations algorithmiques comparables ont le même algorithme sous-jacent ; pour une factorisation u_x, $u_x(i)$ représente le i^{e} argument impliqué (i.e. x_i) [16]. On peut dériver naturellement de ces considérations un modèle de proportion analogique entre « objets calculables » :

Définition 17 (Proportion analogique (non restreinte)). On a $x : y :: z : t$ ssi il existe des factorisations algorithmiques comparables (u_x, u_y, u_z, u_t) de taille n t.q.

$$\forall i \in [\![1, n]\!], (u_y(i), u_z(i)) \in \{(u_x(i), u_t(i)), (u_t(i), u_x(i))\}, \text{ i.e.}$$

$$\forall i \in [\![1, n]\!], (y_i, z_i) \in \{(x_i, t_i), (t_i, x_i)\}.$$

La *complexité* de la proportion est liée à la taille des algorithmes qui en rendent compte.

Dans ce contexte, toutes les définitions proposées précédemment reposent sur des algorithmes très spécifiques, propres à la nature particulière des objets considérés. Par exemple, dans le cas des monoïdes libres, on considère des machines très simples, qui savent effectuer uniquement l'opération de concaténation. La complexité de ces machines simples est caractérisée par le degré.

Dans le cas général, des objets forment une proportion analogique si et seulement si c'est le cas des algorithmes permettant de les produire ; la complexité de

16. Cette notation peu conventionnelle est utilisée de façon à rester en cohérence avec le reste du chapitre.

la proportion est égale à la taille du plus petit algorithme permettant d'en rendre compte. L'introduction de ce niveau d'indirection est par ailleurs aux fondements de la notion de complexité algorithmique (voir par exemple Li & Vitányi (1997) ; Delahaye (1999)).

Cette définition est maintenant très expressive : elle permet de modéliser et d'expliquer des proportions complexes, en particulier celles étudiées par Hofstadter & the Fluid Analogies Research Group (1995). Par exemple, de façon à modéliser $abc : abd :: ijk : ijl$, il « suffit » de poser $u(a_1, a_2, a_3, a_4) = a_1.a_2.(a_3 + a_4)$. On a alors

cf. Sec. 2.1.4, p. 21

$$u(a,b,c,0) = abc, \;\; u(a,b,c,1) = abd, \;\; u(i,j,k,0) = ijk, \;\; u(i,j,k,1) = ijl.$$

La relation entre les lettres a,b et c peut également être exploitée : avec $u(a_1, a_2) = a_1.(a_1 + 1).(a_1 + 2 + a_2)$, on a $u(a,0) = abc$, $u(a,1) = abd$, $u(i,0) = ijk$, $u(i,1) = ijl$. Cela s'applique aussi à des proportions telles que $abc : abd :: mrrjjj : mrrkkk$ ou $abc : abd :: mrrjjj : mrrkkkk$.

Le travail effectué par Hofstadter et ses coauteurs s'appuie sur l'émergence de structures. Nous rejoignons leur discours : dans l'absolu, les représentations ne sont pas fournies, mais émergent de processus. Cependant, nous ne cherchons pas à expliquer comment cette émergence s'opère cognitivement, même s'il est toujours possible de poser des critères généraux (simplicité, symétrie, etc.).

Il est important de noter ici que le modèle peut être rendu plus ou moins expressif selon la restriction opérée sur la nature des algorithmes considérés ; il y a donc un curseur à positionner, qui correspond à un compromis entre l'expressivité et le coût computationnel.

Ce cadre peut être rapproché du modèle proposé par Schmid *et al.* (2003). Dans celui-ci, quatre termes (a, b, c, d) forment une proportion analogique si et seulement si $a = P\sigma_1$, $b = Q\sigma_1$, $c = P\sigma_2$, $d = Q\sigma_2$ où P sont Q des termes sur lesquels s'appliquent des substitutions σ_1, σ_2. Ces termes peuvent contenir des opérateurs non contraints ; il leur est donc également possible de modéliser des situations assez complexes. En posant $u(P, \sigma) = P\sigma$, on remarque que leur modèle s'insère dans notre cadre général ; le curseur est alors situé quelque part entre le modèle de proportions entre éléments d'un semigroupe et le modèle algorithmique très général.

Soulignons enfin que cette approche généralise (d'un point de vue formel) les visions transformationnelles et dérivationnelles de l'analogie ; des schémas tels que $a : f(a) :: b : f(b)$ et $a : b :: f(a) : f(b)$ s'obtiennent en posant $u(f, x) = f(x)$. Dans ce cas, on a bien $u(Id, a) = a : u(f, x) = f(a) :: u(Id, b) = b : u(f, b) = f(b)$ et $u(Id, a) = a : u(Id, b) = b :: u(f, b) = f(a) : u(f, b) = f(b)$.

4.2.5 Le cas des arbres

Les arbres étiquetés sont abondamment utilisés dans les applications de TAL : ils peuvent servir de représentation à des structures syntaxiques, des décomposi-

tions morphologiques, des relations sémantiques, ou des termes d'une logique. Savoir caractériser la notion de proportion entre arbres est donc d'un intérêt majeur pour l'application de notre modèle. Pour l'exprimer, nous conservons l'idée générale reposant sur les notions de factorisation et d'alternance de facteurs. Dans le cas des arbres, les factorisations s'exprimeront à l'aide de *substitutions* et les facteurs alternant seront des *sous-arbres* ; avant de poursuivre, nous introduisons quelques notations et définitions propres au cas des arbres.

Notations et définitions

Les notations qui suivent proviennent ou sont adaptées de Comon *et al.* (1997).

Définition 18 (Arbre). Un *domaine d'arbre* est un langage préfixiel $\mathcal{P}os \subseteq \mathbb{N}_+^\star$ tel que $\forall (u,i) \in (\mathbb{N}_+^\star \times \mathbb{N}_+), ui \in \mathcal{P}os \Rightarrow \forall j \in [\![1,i]\!], uj \in \mathcal{P}os$.

Un *arbre* (ordonné) t sur un *ensemble d'étiquettes* L est une application d'un domaine d'arbre $\mathcal{P}os(t)$ vers L. On désigne par $\mathcal{T}(L)$ l'ensemble des arbres sur L. On distinguera un sous-ensemble $V \subset L$ composé de *variables*.

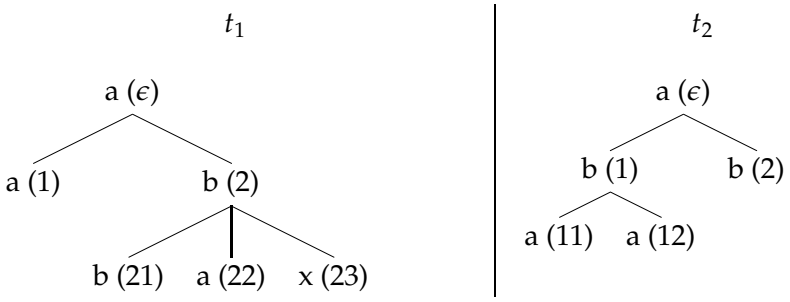


FIG. 4.14 – Deux exemples d'arbres

Tout élément de $\mathcal{P}os(t)$ est appelé une *position* ou un *nœud*. La taille $|t|$ d'un arbre désigne le nombre de ses positions, i.e. $|\mathcal{P}os(t)|$. Une position q est un *enfant* (direct) de p si $q = pi$ pour un $i \in \mathbb{N}_+$. L'ensemble des enfants de p dans t est noté $\mathcal{E}nfants(t,p)$. L'*arité* d'une position p désigne $|\mathcal{E}nfants(t,p)|$. Une *position frontière* ou *feuille* est une position p d'arité 0. L'ensemble des feuilles est noté $\mathcal{FP}os(t)$. Chaque position p telle que $t(p) \in V$ est appelée une *position de variable*. L'ensemble des positions de variables de t est noté $\mathcal{VP}os(t)$ et $\mathcal{V}(t) = t(\mathcal{VP}os(t))$ est l'ensemble des variables apparaissant dans t. L'ensemble des positions étiquetées par $l \in L$ dans t est décrit par $\mathcal{P}(t,l) = \{p \in \mathcal{P}os(t) \mid t(p) = l\}$. La *racine* de t désigne $t(\epsilon)$ et sera notée $r(t)$. Un arbre tel que $\mathcal{P}os(t) = \{\epsilon\}$ est un *arbre racine*. Dans cette section, nous considérons uniquement des arbres dont les variables sont situées sur les feuilles, i.e. $\mathcal{VP}os(t) \subseteq \mathcal{FP}os(t)$.

Définition 19 (Sous-arbre). Un *sous-arbre* $t|_p$ d'un arbre $t \in \mathcal{T}(L)$ à la position $p \in \mathcal{P}os(t)$ est décrit par :

- $\mathcal{P}os(t|_p) = p^{-1}\mathcal{P}os(t)$;
- $\forall q \in \mathcal{P}os(t|_p), t|_p(q) = t(pq)$.

Il est possible d'écrire $\mathcal{P}os(t) = \{\epsilon\} \cup \bigcup_{i=1}^{n} i\mathcal{P}os(t|_i)$; on pourra utiliser la notation $r(t)(t|_1, t|_2, \ldots, t|_n)$ pour désigner un arbre t, où n représente l'arité de ϵ.

Exemple Soit l'ensemble d'étiquettes $L = \{a, b, x\}$, avec $V = \{x\}$ l'ensemble des variables. L'arbre $t_1 \in \mathcal{T}(L)$ défini par

$$\mathcal{P}os(t_1) = \{\epsilon, 1, 2, 21, 22, 23\},$$
$$t_1(\epsilon) = t_1(1) = t_1(22) = a, t_1(2) = t_1(21) = b \text{ et } t_1(23) = x$$

est représenté sur la figure 4.14, dans laquelle les positions apparaissent entre parenthèses.

Définition 20 (Substitution). Une *substitution* (unitaire) est une paire $(v \leftarrow t')$, où $v \in V$ est une variable et $t' \in \mathcal{T}(L)$ un arbre. L'application de la substitution $(v \leftarrow t')$ à un arbre t s'opère en remplaçant chaque feuille de t étiquetée par v par l'arbre t'. Le résultat de cette opération est noté $t \lhd_v t'$, ce qui définit un opérateur binaire $\lhd_v$.

L'application d'une substitution peut être calculée récursivement comme suit :

$$t \lhd_v u = \begin{cases} u \text{ si } t \text{ est un arbre racine et } r(t) = v \\ t \text{ si } t \text{ est un arbre racine et } r(t) \neq v \\ r(t)(t|_1 \lhd_v u, \ldots, t|_n \lhd_v u), \text{ où } n \text{ est l'arité de } \epsilon, \text{ sinon.} \end{cases}$$

Exemple de substitution Appliquer à t_1 la substitution $(x \leftarrow t_2)$ donne le résultat $t_3 = t_1 \lhd_x t_2$ (cf. figure 4.15).

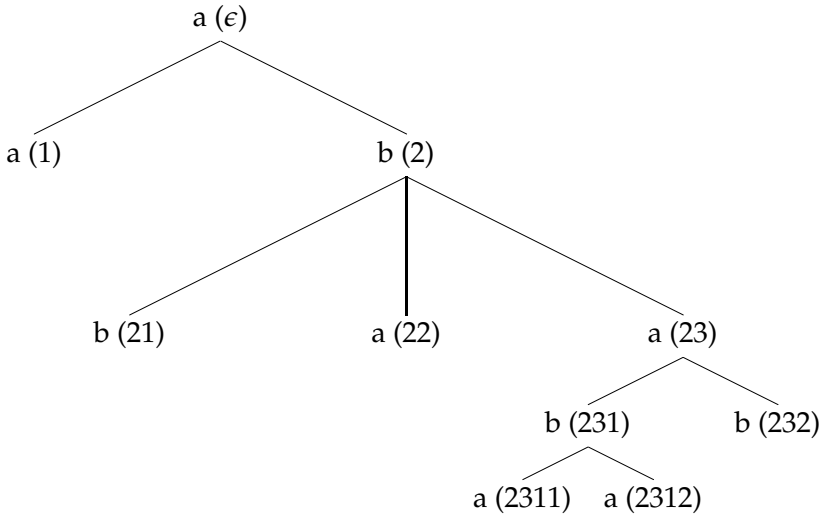


FIG. 4.15 – Le résultat t_3 de la substitution $t_1 \lhd_x t_2$

La notion de factorisation en résulte directement.

Définition 21 (Factorisation). Une *factorisation* d'un arbre $t \in \mathcal{T}(L)$ est définie par une séquence de variables $(v_1, \ldots, v_{n-1})$ et une séquence de sous-arbres $(t_1, \ldots, t_n)$ telles que $t_1 \lhd_{v_1} \ldots \lhd_{v_{n-1}} t_n = t$.

cf. Sec. 4.2.2

La séquence de variables impliquées dans une factorisation correspond au « squelette » de celle-ci ; deux factorisations comparables reposent sur des squelettes identiques, c'est-à-dire les même séquences de variables. Pour une factorisation f_t, $f_t(i)$ représente le i^{e} sous-arbre de la séquence de sous-arbres, c'est-à-dire t_i.

Proportions analogiques entre arbres

La définition de la notion de proportion analogique entre arbres suit immédiatement.

Définition 22 (Proportion analogique (entre arbres)). Pour $(x, y, z, t) \in \mathcal{T}(L)^4$, on a $x : y :: z : t$ si et seulement s'il existe des factorisations d'arbres comparables $(f_x, f_y, f_z, f_t) \in \mathcal{F}(\mathcal{T}(L))^4$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}, \text{ i.e.}$$

$\exists (x_i)_{i\in[\![1,d]\!]}, (y_i)_{i\in[\![1,d]\!]}, (z_i)_{i\in[\![1,d]\!]}, (t_i)_{i\in[\![1,d]\!]}$ t.q.

$$\begin{aligned} x_1 \lhd_{v_1} \dots \lhd_{v_{n-1}} x_n &= x, \\ y_1 \lhd_{v_1} \dots \lhd_{v_{n-1}} y_n &= y, \\ z_1 \lhd_{v_1} \dots \lhd_{v_{n-1}} z_n &= z, \\ t_1 \lhd_{v_1} \dots \lhd_{v_{n-1}} t_n &= t, \end{aligned}$$

$$\text{et } \forall i \in [\![1, d]\!], (y_i, z_i) \in \{(x_i, t_i), (t_i, x_i)\}.$$

Cette définition est une adaptation de la définition 12 dans laquelle les sous-arbres sont agrégés à l'aide de substitutions. En revanche, on ne considère ici non plus un seul opérateur mais un ensemble d'opérateurs (puisque chaque variable induit un opérateur) ; soulignons que nous avions par ailleurs déjà été amené à considérer plusieurs opérateurs (deux) dans le cas des langages.

cf. Sec. 4, p. 76

Il est par ailleurs possible de vérifier que cette notion de proportion couvre les exemples de l'introduction du chapitre, ainsi que celui illustré sur la figure 4.16. La figure 4.17 explicite les factorisations sous-jacentes à cette proportion. Voir également les figures 4.18 et 4.19.

Calcul de proportions entre arbres

Nous présentons dans cette section une méthode approximative permettant de calculer des proportions entre arbres, qui repose sur une conversion des arbres en chaînes.

Conversion des arbres en chaînes Une linéarisation d'arbre est une application injective d'un ensemble d'arbres vers un monoïde libre ; une linéarisation établit une correspondance unique entre un arbre et une chaîne. Nous présentons ici deux linéarisations différentes.

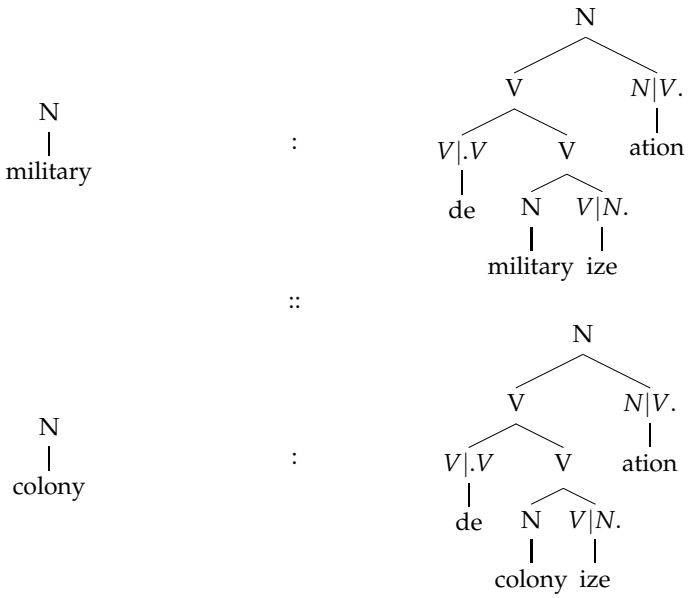


FIG. 4.16 – *military* : *demilitarization* :: *colony* : *decolonization*

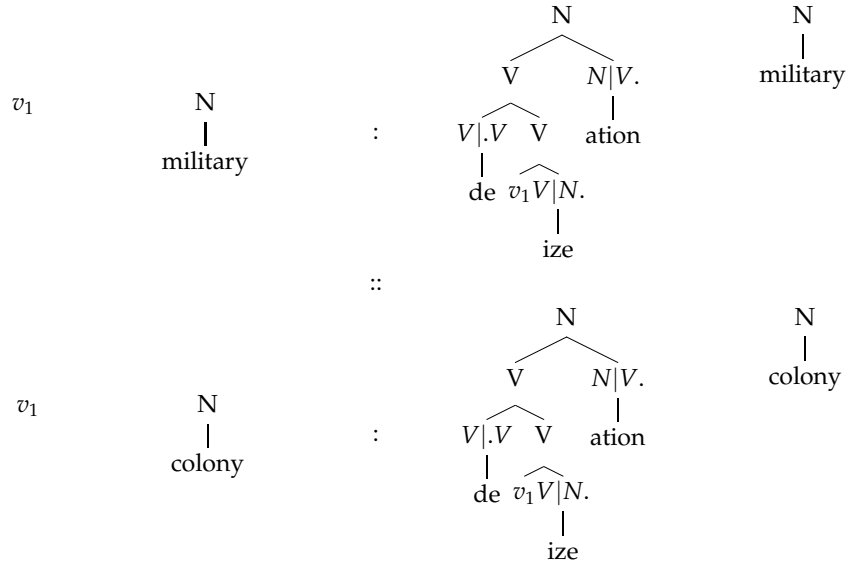


FIG. 4.17 – *military* : *demilitarization* :: *colony* : *decolonization* (factorisations)

Définition 23 (Expressions parenthésées). Soit L un ensemble d'étiquettes et $'('$,$')'$ deux symboles absents de L. $\overline{L}$ désigne $L \cup \{(,)\}$. L'*expression parenthésée* d'un arbre t est un mot $p(t) \in \overline{L}^\star$ défini par la récurrence [17] :

$$p(t) = (.r(t).p(t|_1).\ldots.p(t|_n).),$$

où n est l'arité de ϵ.

17. La concaténation est explicitement notée pour rendre la lecture plus claire.

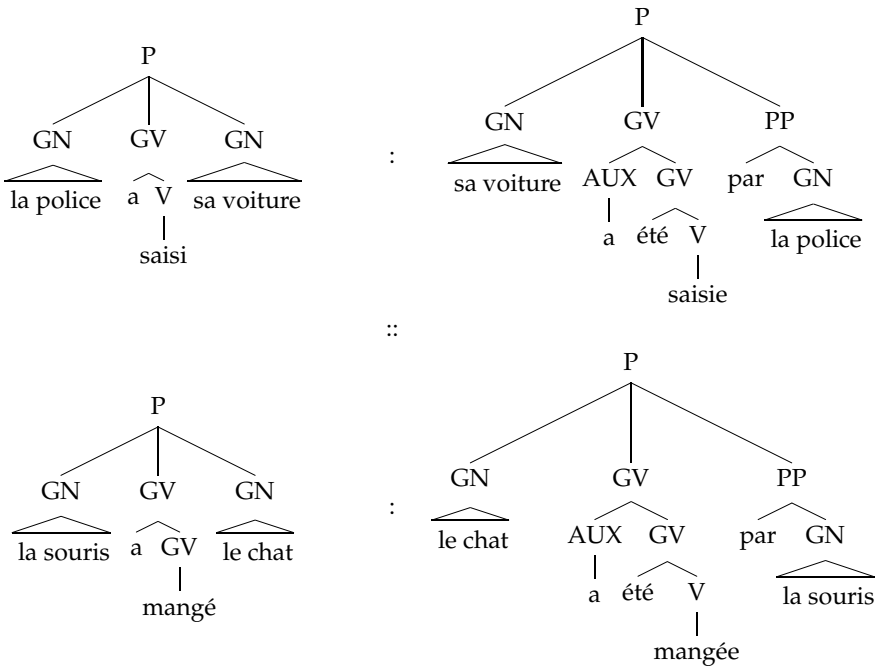


FIG. 4.18 – Proportion actif/passif

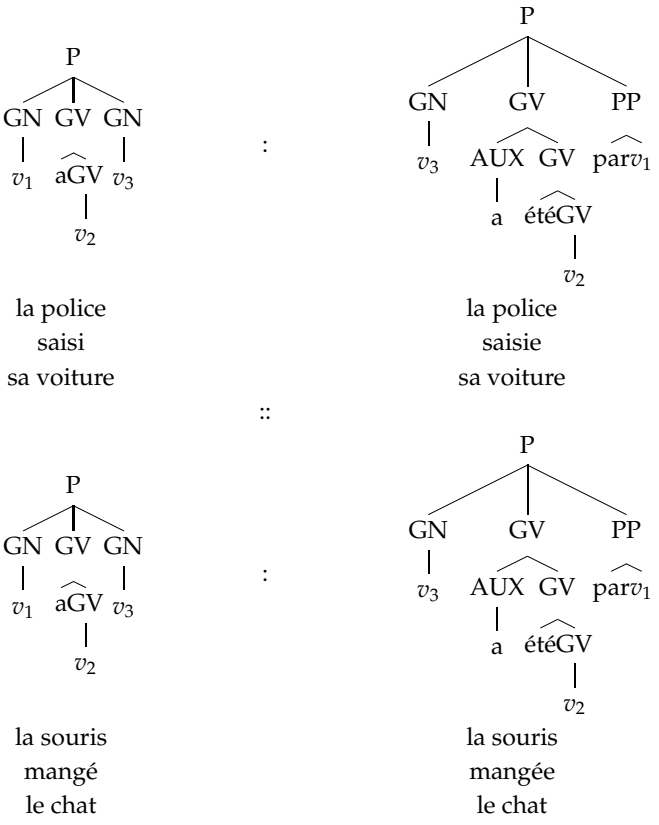


FIG. 4.19 – Proportion actif/passif (factorisation)

Exemple $p(t_3) = (a(a)(b(b)(a)(a(b(a)(a))b)))$. *cf. Sec. 4.2.5, p. 111*

Définition 24 (Linéarisation fondée sur l'arité). La *linéarisation fondée sur l'arité* d'un arbre t est le mot $ab(t) \in (L \times \mathbb{N})^\star$ défini par la récurrence :

$$ab(t) = (r(t), n)ab(t|_1) \dots ab(t|_n),$$

où n est l'arité d'ϵ.

Exemple $ab(t_3) = (a,2)(a,0)(b,3)(b,0)(a,0)(a,2)(b,2)(a,0)(a,0)(b,0)$.

Par construction, ces applications sont injectives.

Un solveur approximatif Nous fournissons maintenant des résultats explicitant les liens entre les proportions entre arbres et les proportions entre leurs linéarisations (vues comme des chaînes), dans le cas particulier où les variables n'apparaissent qu'une seule fois dans les sous-arbres, et dans le même ordre relatif, c'est-à-dire, pour $\alpha \in \{x, y, z, t\}$ et les factorisations $(\alpha_i)_{i \in [\![1,n]\!]}$:

- $\forall i \in [\![1, n]\!], \forall v \in \mathcal{V}(\alpha_i), |\mathcal{P}(\alpha_i, v)| = 1$;
- $\forall i \in [\![1, n]\!], \forall v, w \in \mathcal{V}(\alpha_i)$,

$$\mathcal{P}(x_i, v) < \mathcal{P}(x_i, w) \Leftrightarrow \mathcal{P}(t_i, v) < \mathcal{P}(t_i, w).$$

Une proportion (entre arbres) vérifiant ces conditions sera notée $x : y \overset{C}{::} z : t$.

Proposition 14. Pour $(x, y, z, t) \in (\mathcal{T}(L))^4$:

$$x : y \overset{C}{::} z : t \Rightarrow p(x) : p(y) :: p(z) : p(t).$$

Démonstration. Par récurrence sur le degré de la proportion. Soit quatre termes $(x, y, z, t) \in (\mathcal{T}(L))^4$ tels que $x : y \overset{C}{::} z : t$. Par définition, il existe des factorisations comparables $(f_x, f_y, f_z, f_t) \in \mathcal{F}(\mathcal{T}(L))^4$ telles que

$$\forall i \in [\![1, d]\!], (f_y(i), f_z(i)) \in \{(f_x(i), f_t(i)), (f_t(i), f_x(i))\}.$$

Le résultat est trivialement vérifié pour $d = 1$. On fait maintenant l'hypothèse que le résultat est vrai pour toute proportion de degré $d - 1$. On fait également l'hypothèse que pour toute proportion de degré $d - 1$, pour toute variable v apparaissant une seule fois dans $\alpha' \in \{x', y', z', t'\}$, si $p(\alpha') = u_{\alpha'_1} p(v) u_{\alpha'_2}$, alors $u_{x'_1} : u_{y'_1} :: u_{z'_1} : u_{t'_1}$ et $u_{x'_2} : u_{y'_2} :: u_{z'_2} : u_{t'_2}$ (ce qui est vérifié pour $d = 1$ puisque les proportions sont alors atomiques).

Pour $\alpha \in \{x, y, z, t\}$, on pose $\alpha' = \alpha_1 \lhd_{v_1} \dots \lhd_{v_{d-2}} \alpha_{d-1}$ et $v = v_{d-1}$; on a alors $\alpha = \alpha' \lhd_v \alpha_d$. Par hypothèse de récurrence, la proportion $p(x') : p(y') :: p(z') : p(t')$ est vérifiée. En outre, puisque v apparaît une seule fois dans chaque sous-arbre, on peut écrire $p(\alpha') = u_{\alpha'_1} p(v) u_{\alpha'_2}$ pour $\alpha \in \{x, y, z, t\}$, donc $p(\alpha) = u_{\alpha'_1} p(\alpha_d) u_{\alpha'_2}$, ce qui donne $p(x) : p(y) :: p(z) : p(t)$ par concaténation des trois proportions

$$u_{x'_1} : u_{y'_1} :: u_{z'_1} : u_{t'_1}, p(x_d) : p(y_d) :: p(z_d) : p(t_d) \text{ et } u_{x'_2} : u_{y'_2} :: u_{z'_2} : u_{t'_2}.$$

On doit maintenant vérifier que la propriété ajoutée aux hypothèses de récurrence est également vérifiée pour d, i.e. pour toute variable w, si $p(\alpha) = u_{\alpha_1} p(w) u_{\alpha_2}$, alors $u_{x_1} : u_{y_1} :: u_{z_1} : u_{t_1}$ et $u_{x_2} : u_{y_2} :: u_{z_2} : u_{t_2}$. Remarquons tout d'abord que w apparaît ou bien dans α', ou bien dans α_d. Dans le premier cas, le résultat est vérifiée par hypothèse de récurrence. Si w est dans α_d, il est suffisant d'observer que les variables dans x_d et t_d apparaissent une seule fois, et dans le même ordre relatif.

□

Proposition 15. Pour $(x, y, z, t) \in (\mathcal{T}(L))^4$:

$$x : y \overset{C_1}{::} z : t \Rightarrow ab(x) : ab(y) :: ab(z) : ab(t)$$

Démonstration. La démonstration s'obtient directement en remplaçant $p(w)$ par $ab(w)$ dans la preuve de la proposition 14. □

Ces résultats légitiment en partie l'utilisation du solveur d'équations utilisé dans le cas des chaînes [18]. Puisque l'implication n'est vérifiée que dans un seul sens, cette utilisation correspond à une approximation par filtrage. En effet, le résultat d'implication nous permet de filtrer les cas dont on est sûrs qu'ils ne correspondent pas à une proportion, les autres étant conservés par défaut. Il est par ailleurs possible d'additionner ces filtres de manière à se rapprocher de l'exactitude ; par exemple, on peut coupler les deux linéarisations évoquées. Signalons enfin qu'un travail concernant un algorithme exact est en cours.

4.3 Conclusion

Dans ce chapitre, nous avons cherché à modéliser la notion de proportion analogique formelle, avec pour objectif l'identification de proportions entre représentations courantes d'objets linguistiques. Pour cela, nous sommes partis du modèle de proportions entre chaînes initialement proposé par Yvon (2003). Nous avons pu l'étudier en détail et proposer en particulier des mesures qualifiant la qualité d'une proportion analogique. Nous avons montré que ce modèle s'étend, d'un point de vue définitionnel, au cas des semigroupes et des magmas, fournissant un cadre algébrique très général. Plusieurs résultats appuient la cohérence globale du modèle proposé.

Ce cadre s'instancie pour donner un sens à la notion de proportion analogique entre des objets structurés tels que les structures de traits, les langages finis ou les arbres. Cependant, alors que le modèle de proportions entre chaînes conduit à une caractérisation calculatoire précise, la présence de quantificateurs existentiels dans la définition limite l'application directe du modèle dans le cas général. Nous disposons pour l'instant de quelques résultats qui fournissent des simplifications dans plusieurs situations ; c'est le cas, en particulier, des structures de traits. En

18. Lepage (1999) utilise également un tel type de linéarisation pour travailler sur les arbres ; en revanche, aucune justification n'est fournie.

ce qui concerne les arbres, un solveur approximatif peut être dérivé du solveur s'appliquant aux chaînes.

CHAPITRE

5 Validations expérimentales

J'ai beau y réfléchir, je n'vois pas d'solution.

— Didier Wampas, *Les bottes rouges*

Sommaire du chapitre

NOUS avons étudié dans le chapitre 3 une méthode d'apprentissage automatique, reposant sur la notion de proportion analogique et adaptée à l'analyse de changement de niveau de représentation linguistique. Cette méthode présente les propriétés de symétrie attendues ; elle est également en mesure de manipuler des objets structurés dans l'espace d'entrée comme dans l'espace de sortie, dès lors que des procédures de recherche de triplets et de résolution d'équation analogique sont disponibles. Des modèles formels de proportions analogiques ont été proposés dans le chapitre 4, qui permettent de donner un sens à la notion de proportion relativement à de nombreuses structures algébriques telles que les semigroupes, les monoïdes, les groupes, les treillis, etc. En outre, pour un certain nombre de structures particulières, ces modèles donnent lieu à des algorithmes efficaces implantant les procédures requises.

Nous disposons désormais de tous les ingrédients nécessaires à un apprentissage automatique de données linguistiques. Dans ce chapitre, nous évaluons les capacités de généralisation de notre méthode sur un certain nombre d'applications de TAL s'exprimant de manière naturelle comme un changement de niveau de représentation. En particulier, nous étudions les tâches de transcription orthographique/phonétique (cf. section 5.2), d'analyse flexionnelle (cf. section 5.3) et d'analyse dérivationnelle (cf. section 5.4).

La méthodologie générale adoptée pour ces expériences est décrite dans la section 5.1. Nous y présentons également des réponses aux limitations soulevées précédemment concernant l'algorithme APPA, à savoir (i) l'agrégation et le filtrage de solutions, (ii) la recherche des triplets. Nous comparons les résultats obtenus au système TIMBL [1] (Daelemans *et al.*, 2004). De façon à effectuer les expériences décrites, nous avons développé un logiciel, baptisé ALANIS (*A Learning by ANalogy Inferencer for Structured data*). Celui-ci fournit un cadre générique permettant de travailler avec une large gammes de structures ; il repose sur la bibliothèque VAUCANSON (Lombardy *et al.*, 2004) qui offre un polymorphisme sans perte d'efficacité [2] (Régis-Gianas & Poss, 2003 ; Burrus *et al.*, 2003). Le logiciel ALANIS est décrit plus en détail dans l'annexe B.

cf. Sec. 3.1.2, p. 45

5.1 Méthodologie générale

Dans cette section, nous précisons des points sur la méthodologie générale adoptée, commune à toutes les expériences effectuées.

5.1.1 Évaluation des performances d'un système d'apprentissage

L'objectif d'un système d'apprentissage automatique (supervisé) est, à partir d'une base d'apprentissage d'exemples analysés, de construire un apprenti en mesure d'analyser un objet non encore rencontré (i.e. absent de la base d'appren-

1. Disponible depuis `http://ilk.uvt.nl/timbl/`.
2. Il est question de polymorphisme statique.

tissage). La démarche dominante pour évaluer un tel système et le comparer à d'autres consiste à estimer sa performance en terme de *justesse des prédictions*. Cette justesse est caractérisée par un ensemble de quantités mesurant la capacité du système à effectuer une prédiction correcte sur un objet non connu.

En pratique, on dispose d'un ensemble restreint de données analysées et les analyses des objets réellement inconnus ne sont pas disponibles. La démarche habituellement adoptée pour effectuer néanmoins une évaluation consiste à séparer la base d'exemples analysés en deux ensembles formant respectivement *une base d'apprentissage* et une *base de tests*. Un système d'apprentissage utilisera la base d'apprentissage pour construire un apprenti ; cet apprenti analysera ensuite les exemples de la base de tests, et les analyses qu'il fournira seront comparées aux analyses réelles [3].

Les estimations impliquées dans une évaluation sont donc obtenues à l'aide de mesures effectuées sur les données de la base de tests ; les résultats peuvent varier significativement selon la méthode de séparation adoptée (taille des bases, choix des exemples, etc.). Sur les méthodes d'évaluation et de comparaison d'algorithmes d'apprentissage, voir par exemple Mitchell (1997, chap. 5), Cornuéjols & Miclet (2002, sec. 3.4) et Dietterich (1997).

L'approche adoptée implique m étapes (cf. algorithme 6). À chaque étape, n exemples sont tirés aléatoirement et retirés de la base de données pour former la base de tests ; l'algorithme est entraîné sur les exemples restants (la base d'apprentissage) puis testé sur ces n exemples. Les résultats sont ensuite moyennés sur ces m étapes. Dans toutes les expériences effectuées, nous avons posé $m = 10$ et

entrée : BD = base des données
entrée : n = nombre d'essais
entrée : m = taille de la base de tests
sortie : estimation des prédictions
for $i = 0; i < m$ **do**
  $BA \longleftarrow BD$
  $BT \longleftarrow \emptyset$
  for $k = 0; k < n$ **do**
    $x \longleftarrow$ exemple tiré aléatoirement de BA
    $BT \longleftarrow BT \cup \{x\}$
    $BA \longleftarrow BA \setminus \{x\}$
  end
  apprenti = construire_apprenti(BA)
  p_i = evalue_apprenti(BT)
end
return $moyenne((p_i)_{i \in [\![1,m]\!]})$

Algorithme 6 : evalue_methode_apprentissage

3. Il est également possible de considérer un jeu de données supplémentaire dont l'objectif est de permettre l'optimisation ou l'ajustement de paramètres entre la phase d'apprentissage et la phase de tests ; on parlera alors de base de développement.

$n = 1000$. Cette approche est justifiée d'un point de vue statistique car le nombre de données disponibles est élevé (bases de centaines de milliers d'exemples). D'une part, le fait que le nombre de données disponibles est élevé assure que la suppression de 1000 exemples ne perturbe pas l'apprentissage ; d'autre part, un échantillon de taille 1000 est suffisant pour estimer précisément les capacités de généralisation d'un système d'apprentissage. Pour que les résultats soient significatifs, l'évaluation porte généralement sur différents jeux de données ; nous avons effectué des expériences sur plusieurs lexiques correspondant à des langues différentes.

Précision, rappel et F-score De façon à rendre compte de la justesse de prédiction, un certain nombre de mesures peuvent être prises en considération ; les plus courantes sont la *précision* et le *rappel*, mesures introduites en premier lieu en Recherche d'Information (van Rijsbergen, 1979).

Dans les tâches étudiées, à une entrée x_i correspondent potentiellement plusieurs sorties : y_i est donc un ensemble. De même, les systèmes d'apprentissage pourront fournir plusieurs solutions pour une entrée donnée : $f(x_i)$ est donc également un ensemble dans le cas général.

Dans ce cadre, la précision p_i associée à un couple $(y_i, f(x_i))$ correspond au nombre de sorties correctes proposées rapporté au nombre total de sorties proposées, i.e. :

$$p_i = \frac{|f(x_i) \cap y_i|}{|f(x_i)|}.$$

Le rappel r_i correspond au nombre de sorties correctes proposées rapporté au nombre total de sorties attendues [4] :

$$r_i = \frac{|f(x_i) \cap y_i|}{|y_i|}.$$

La précision globale s'obtient en moyennant la précision sur chaque exemple :

$$p = \frac{1}{n} \sum_{i=1}^{n} p_i.$$

Si on pondère les exemples par le nombre de sorties attendues, alors la précision globale devient :

$$p = \frac{\sum_{i=1}^{n} |y_i| p_i}{\sum_{i=1}^{n} |y_i|}.$$

On parlera de *micro-précision* dans le premier cas, et de *macro-précision* dans le deuxième ; les mêmes remarques s'appliquent au cas du rappel. Alors que les macro-mesures donnent plus de poids aux exemples aux sorties multiples, les micro-mesures traitent tous les exemples de manière identique.

4. Si $|f(x_i)| = |y_i| = 0$, on pose $p_i = r_i = 1$; si $|f(x_i)| = 0$ et $|y_i| \neq 0$, on pose $p_i = 1$, $r_i = 0$, si $|f(x_i)| \neq 0$ et $|y_i| = 0$, $p_i = 0$, $r_i = 0$.

Pour comparer deux systèmes à l'aide d'une seule mesure, on agrège habituellement le rappel et la précision selon le *f-score* F_β, défini par [5] :

$$F_\beta = \frac{(\beta^2+1)pr}{\beta^2 p + r}.$$

Cette mesure permet d'agréger le rappel et la précision tout en autorisant à favoriser le rappel ($\beta < 1$) ou la précision ($\beta > 1$). Sans connaissance spécifique supplémentaire, on choisit généralement $\beta = 1$, i.e.

$$F = \frac{2pr}{p+r}.$$

5.1.2 Stratégies de filtrage et de pondération

L'algorithme APPA propose potentiellement plusieurs solutions pour une entrée donnée (plusieurs triplets fournis par la recherche de triplets, plusieurs solutions possibles à une équation analogique). Cette propriété est bienvenue pour la plupart des tâches que l'on désire traiter puisqu'impliquant de l'ambiguïté. Ce comportement nécessite de disposer de mécanismes d'agrégation et de filtrage de solutions. En effet, il n'est pas nécessairement judicieux de conserver toutes les solutions proposées ; seules les meilleures doivent être retenues. Par ailleurs, savoir pondérer les solutions constitue une propriété très appréciable du modèle : les probabilités des sorties attendues peuvent être inégales, et il est bienvenu de pouvoir en rendre compte ; il est également possible d'envisager de réutiliser les scores fournis par le système à l'aide d'autres mécanismes (méta-apprentissage, etc.).

Deux critères sont utilisés pour mesurer la qualité d'une solution : le degré des proportions et la taille des paradigmes impliqués. Formellement, le poids associé à une proposition o_x pour une entrée i_x est donné par :

$$w(i_x, o_x) = \sum_{(x_i,x_j,x_k)\in BA} 2^{-(D(i(x_i) : i(x_j) :: i(x_k) : i_x)+D(o(x_i) : o(x_j) :: o(x_k) : o_x))},$$

où $D(x : y :: z : t)$ désigne le degré [6] de la proportion $x : y :: z : t$. De cette façon, le degré est pris en compte pour tous les triplets et équations analogiques considérés. La taille des paradigmes est représentée par la sommation effectuée sur tous les triplets ; le fait de trouver de nombreuses proportions supportant une solution confirme l'existence d'un paradigme de taille importante.

Cette mesure associe un poids à chaque solution fournie et permet donc de les classer. Dans la suite, nous noterons $S(i_x) = \{s_1, \dots s_n\}$ l'ensemble ordonné des solutions proposées pour i_x ($w(i_x, s_i) \geq w(i_x, s_j)$ pour $i < j$). À partir de ce classement, plusieurs stratégies de filtrage peuvent être envisagées.

5. Le F-score correspond en réalité à une moyenne harmonique du rappel et de la précision : $\frac{1}{F_\beta} = \alpha\frac{1}{p} + (1-\alpha)\frac{1}{r}$ avec $\beta^2 = \frac{1-\alpha}{\alpha}$.

6. Si la proportion $x : y :: z : t$ n'est pas vérifiée, on pose $D(x : y :: z : t) = \infty$.

Stratégies de seuillage Deux techniques simples, *Threshold* et *KBest*, reposent sur l'introduction d'un seuil. *Threshold* fixe un seuil minimal t pour les solutions considérées, i.e.

$$Threshold(S(i_x), t) = \{s \in S(i_x) | w(i_x, s) > t\}.$$

Avec *KBest*, le seuil concerne le nombre de solutions retenues : on ne considère que les k meilleures solutions :

$$KBest(S(i_x), k) = \{s_1, \ldots, s_k\}.$$

Les seuils fournissant les meilleurs résultats peuvent être fixés a priori par les besoins de l'application, ou optimisés à l'aide d'un ensemble de développement. Fixer un seuil identique pour tous les exemples considérés n'est pas optimal ; en effet, le nombre de solutions attendues varie en fonction des exemples. En revanche, en observant la variation des performances de généralisation en fonction de la variation de ces seuils, il est possible de mettre en évidence le fait que le classement adopté est valide ; autrement dit, les solutions auxquelles on a affecté des poids élevés sont effectivement le plus souvent des solutions correctes.

Augmentation d'entropie De façon à trouver dynamiquement et automatiquement le nombre correct de solutions à fournir, nous proposons la méthode AE (augmentation d'entropie). Cette méthode essaie d'isoler les solutions qui forment le « peloton de tête ». Deux exemples de sorties fournies par le système sont illustrés sur la figure 5.1. En observant les poids associés aux solutions, on « voit » que dans le premier exemple (*afflict*), la première solution (*@flIkt*) se détache nettement et que c'est elle seule qu'il faut retenir ; pour le deuxième exemple (*aspersion*), deux solutions (*@sp3SH* et *sp3SH*) semblent devoir être conservées.

Le groupe des solutions formant le peloton de tête est caractérisé par une certaine uniformité relativement aux pondérations. La procédure AE considère la meilleure solution et ajoute itérativement les solutions suivantes tant que cet ajout ne crée pas une cassure brutale d'uniformité de poids dans l'ensemble des solutions retenues (signifiant une solution de poids significativement inférieur à ceux des premières solutions). L'entropie constitue un moyen naturel pour mesurer l'uniformité d'un ensemble[7]. Dans la méthode AE (algorithme 7), on assimile les poids normalisés des solutions à des probabilités ; l'entropie de l'ensemble des solutions est calculée à chaque étape, et l'algorithme se termine lorsque l'ajout d'une solution conduit à une diminution d'entropie.

Moyenne Dans cette stratégie de filtrage (notée *Mean*), on ne conserve que les solutions dont le poids est supérieur au poids moyen de l'ensemble des solutions.

7. Rappelons que l'entropie H d'un ensemble $\{s_1, \ldots, s_i\}$ dont les éléments ont pour probabilités respectives les quantités $\{p_1, \ldots, p_i\}$ est définie par : $H = -\sum_{k=1}^{i} p_i ln_2(p_i)$.

```
<wordform val="afflict" />
<pronunciation val="@flIkt" />
<solution val="@flIkt" weight="7.17188" />
<solution val="@flIktI" weight="0.359375" />
<solution val="@flIkId" weight="0.1875" />
<solution val="@flIk@t" weight="0.125" />

<wordform val="aspersion" />
<pronunciation val="@sp3SH" />
<pronunciation val="{sp3SH" />
<solution val="@sp3SH" weight="9.5625" />
<solution val="{sp3SH" weight="9.5625" />
<solution val="@sp3S7n" weight="0.5" />
<solution val="@sp3Sj@n" weight="0.5" />
<solution val="{sp3S7n" weight="0.5" />
<solution val="{sp3Sj@n" weight="0.5" />
<solution val="{sp3t7n" weight="0.25" />
<solution val="{sp3tj@n" weight="0.25" />
```

FIG. 5.1 – Exemples (simplifiés) de sorties pondérées fournies par ALANIS

entrée : Un ensemble ordonné de solutions $S(i_x)$
sortie : Un sous-ensemble $AE(S(i_x)) \subseteq S(i_x)$
begin
  $AE \longleftarrow \emptyset$
  $E_0 \longleftarrow 0$
  for $j = 1; j < n$ **do**
    $W_j = \sum_{k=1;k\leq j} w(i_x, s_k)$
    $E_j = -\sum_{k=1;k\leq j} \frac{w(i_x,s_k)}{W_j} log(\frac{w(i_x,s_k)}{W_j})$
    if $E_{j+1} < E_j$ **then**
      break
    else
      $AE \longleftarrow s_j$
    end
  end
end
return AE

Algorithme 7 : Augmentation d'entropie

Stratégie du singe La stratégie du singe (notée *Monkey*) consiste à fournir les k meilleures solutions, où k est le nombre de réponses effectivement attendues ; elle est pratiquement optimale. En revanche, elle nécessite de connaître à l'avance ce nombre ; elle n'est donc pas totalement automatique. L'intérêt d'une telle stratégie

est de pouvoir mesurer la qualité relative d'autres stratégies (totalement automatiques, elles), telles que *AE* et *Mean*. En outre, elle fournit une indication sur ce que l'on peut attendre de l'algorithme APPA ; puisque les stratégies de filtrage sont définies totalement indépendamment de celui-ci, il est nécessaire de connaître la contribution respective de chacune des procédures.

5.1.3 Recherche efficace de triplets

La deuxième limitation identifiée à la section 3.2.3 concerne la recherche des triplets ; une recherche aveugle conduit à l'exploration de l'espace BA_X^3, rendant prohibitive l'application de l'algorithme au traitement de larges bases de données. Nous proposons dans cette section des réponses à ce problème [8].

Supposons que l'on puisse associer une variable catégorielle à chaque exemple de la base d'apprentissage, sans pour l'instant la spécifier davantage. Dans ces conditions, la base d'apprentissage peut se réécrire

$$BA = \{(i(x_1), o(x_1), c(x_i)), \ldots, (i(x_n), o(x_n), c(x_n))\}.$$

L'hypothèse analogique repose sur des proportions valides à tous les niveaux. Pour une entrée i_x, on cherche ainsi une solution o_x telle que $i(x_i) : i(x_j) :: i(x_k) : i_x$ et $o(x_i) : o(x_j) :: o(x_k) : o_x$; en outre, si la notion entourant la variable catégorielle est fondée, la proportion doit maintenant être également valide sur celle-ci, i.e. $c(x_i) : c(x_j) :: c(x_k) : c_x$. Sans information supplémentaire sur la nature des catégories, il est légitime de ne considérer que les proportions atomiques, de la forme $a : a :: b : b$ ou $a : b :: a : b$. Ainsi, quelle que soit la catégorie associée à l'objet à analyser, les triplets à considérer sont toujours constitués d'un couple d'éléments appartenant à une même catégorie et d'un élément a priori quelconque. Si les catégories sont nombreuses et relativement petites, cette remarque permet de diminuer considérablement le nombre de triplets à examiner. En effet, en notant n_k la taille du k^{e} paradigme, on a $|BA_X| = \sum n_k$, et les $|BA_X|^3 = |BA_X|(\sum n_k)^2$ triplets initiaux laissent désormais place à $|BA_X|(\sum n_k^2)$ triplets. Le fait d'avoir à vérifier une proportion sur une variable catégorielle est une contrainte forte ; la méthode proposée revient donc en réalité a traiter cette contrainte forte en premier lieu.

De tels regroupements d'exemples en catégories sont donc bénéfiques ; comment peuvent-ils être construits ? Pour répondre à cette question, on observe tout d'abord que les exemples appartenant à la même catégorie sont susceptibles d'être impliqués dans une même proportion ; il paraît donc légitime de regrouper les exemples qui sont de bons candidats aux postes de premier et deuxième termes d'une proportion analogique (cf. figure 5.2), ce qui constituera une première étape. Des connaissances propres à l'application peuvent être exploitées pour guider ces regroupements ; par exemple, dans une tâche d'analyse flexionnelle, il est possible

8. Signalons que les méthodes envisagées ont pour objectif de rendre la recherche de triplets plus efficace en terme de temps de calcul. Le matériel disponible actuellement permet par ailleurs de stocker sans problème particulier des lexiques contenant des centaines de milliers d'entrées ; la question de la réduction de l'espace de stockage des données ne sera donc pas traitée.

de regrouper les exemples en familles lexématiques, c'est-à-dire de former des paradigmes flexionnels. D'autres types de regroupement peuvent également être envisagés.

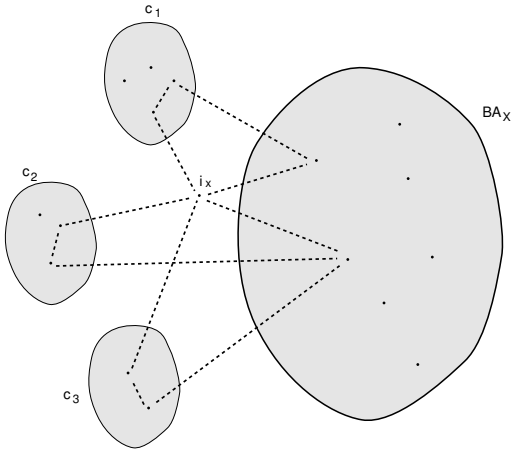


FIG. 5.2 – Regroupement d'exemples en « catégories »

La deuxième étape exploite la redondance naturelle des données linguistiques. Dans ce contexte, les catégories correspondent à des paradigmes linguistiques. L'idée sous-jacente est que les paradigmes sont très redondants et souvent interchangeables : un paradigme peut généralement être remplacé par de nombreux autres. Cette idée a été formalisée à l'aide de la notion de support analogique : seule une partie réduite des données est nécessaire à la reconstruction de celles-ci. Plusieurs méthodes peuvent être envisagées pour construire un support explicitement : techniques de type incrémental, décrémental, etc. ; de telles méthodes ont été précédemment invoquées dans le cas de l'apprentissage paresseux (sur les méthodes d'édition et de condensation dans ce contexte, voir Dasarathy (1990) ; Dasarathy *et al.* (2000) ; Wilson & Martinez (2000)). En réalité, lorsque les données sont suffisamment redondantes, il est possible d'éviter la construction explicite du support. C'est le cas en particulier lorsque, pour la plupart des objets à analyser, de nombreux triplets sont trouvés dans la base d'apprentissage. Dans ce cas, l'approche suivie consiste à sélectionner aléatoirement un certain nombre de paradigmes ; seuls les paradigmes sélectionnés sont examinés. Cette construction correspond à une approximation dynamique du support. Cette méthode est à la fois très simple (elle ne repose pas sur de lourds pré-traitements) et très souple (aucune décision irréversible n'est effectuée). Le nombre de paradigmes sélectionnés est un paramètre de la méthode dont nous étudierons l'influence.

cf. Sec. 3.3.2, p. 58

cf. Sec. 3.2.3, p. 57

5.2 Prononciation

5.2.1 Présentation de la tâche

Dans cette tâche, il s'agit d'associer à toute forme graphique un ensemble de séquences de phonèmes correspondant aux prononciations possibles de la forme.

Les systèmes de synthèse et de reconnaissance vocale, ainsi que les correcteurs orthographiques, nécessitent naturellement ce genre d'analyse (sur cette tâche, voir par exemple Yvon (1996) et Marchand & Damper (2000)). Un exemple de paire d'entrée/sortie pour cette tâche est :

entrée=*infinity*;
sortie= { ɪnfɪnɪtɪ ; ɪnfɪnətɪ }

Nous considérons donc une version simplifiée de la tâche de prononciation, dans laquelle seule importe en sortie la séquence de phonèmes ; en particulier, nous n'apprenons pas la segmentation en syllabes, ou l'accentuation. Selon les langues, la tâche de prononciation peut présenter de l'ambiguïté ; celle-ci est quantifiée plus précisément dans ce qui suit.

5.2.2 Données

Les données utilisées pour effectuer nos tests proviennent du lexique CELEX (Burnage, 1990), couvrant les langues anglaise, néerlandaise et allemande. Pour chacun des lexiques, nous détaillons dans le tableau 5.1 le nombre d'entrées, le nombre d'entrées ambiguës (plus d'une prononciation), le pourcentage d'entrées ambiguës et le nombre moyen de prononciations par entrée. On observe clairement une forte ambiguïté pour l'anglais, et une ambiguïté légère pour l'allemand et le néerlandais. Il est important de noter que cette ambiguïté n'est pas toujours

Corpus	Langue	Nombre d'entrées	Nombre d'entrées ambiguës	Pourcentage d'entrées ambiguës	Nombre moyen de prononciations
CELEX	Anglais	89412	41343	46,24%	1,66
	Néerlandais	324249	12365	3,81%	1,00
	Allemand	342452	16069	4,69%	1,03

TAB. 5.1 – Propriétés des données, tâche de prononciation

prise en compte à sa juste mesure ; autrement dit, il est courant (i) de l'occulter complètement, (ii) de la supprimer en conservant uniquement les prononciations « canoniques » ou les plus fréquentes. Nous considérons au contraire que les ambiguïtés sont inhérentes aux données linguistiques et qu'elles doivent être traitées en tant que telles.

Chaque entrée donne lieu à une ou plusieurs associations entre une forme et une prononciation (ainsi qu'un lexème). Chaque forme est codée par une séquence de symboles `ascii`, correspondant à un alphabet appelé DISC (DIstinct Single Characters), propre au corpus CELEX ; dans cet alphabet, chaque phonème est associé de manière biunivoque à un symbole `ascii`. L'exemple évoqué prendra la forme des deux lignes suivantes (`432` représente le numéro identifiant le lexème) :

```
infinity:432:InfInItI
infinity:432:InfIn@tI
```

5.2.3 Formalisation et pré-traitements

Formalisation nécessaire à APPA Dans le chapitre 4, la notion de proportion analogique est définie relativement à une structure algébrique. Pour pouvoir appliquer l'algorithme APPA, nous devons donc nous placer dans un certain contexte algébrique, c'est-à-dire répondre à la question : « dans la tâche étudiée, à quelles structures algébriques appartiennent les objets considérés ? »

Dans cette tâche, l'entrée est une séquence de symboles graphiques et la sortie une séquence de phonèmes. Pour l'espace d'entrée, la structure algébrique naturelle est le monoïde libre sur l'alphabet des symboles graphiques. Pour l'espace de sortie, le monoïde libre sur l'alphabet des phonèmes est également approprié.

La recherche des triplets effectuée par ALANIS demande de définir des regroupements d'exemples. Dans cette tâche, les entrées sont regroupées par famille flexionnelle (même identifiant de lexème).

Pré-traitement nécessaire à TIMBL (alignement des données) Utiliser des méthodes de classification pour traiter le problème de l'apprentissage de données séquentielles nécessite une reformulation des données. La méthode de fenêtrage, exposée dans la section 3.1.2, opère une telle reformulation. Elle nécessite toutefois un alignement préalable des séquences d'entrée et de sortie.

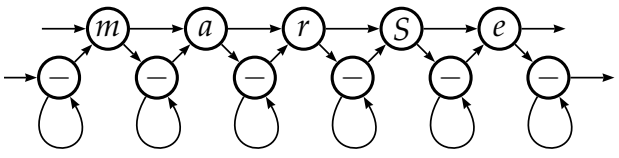
cf. Sec. 3.1.2, p. 45

Dans la suite, nous étudions cinq méthodes d'alignement qui n'utilisent aucune connaissance spécifique sur les données. La première (resp. 2e, 3e, 4e) méthode consiste à effectuer aveuglément un alignement sur la gauche (resp. sur la droite, sur le centre, aléatoirement) ; cet alignement est illustré sur la figure 5.3.

graphème	m	a	r	c	h	e	r
phonème	m	a	r	ʃ	e	_	_

FIG. 5.3 – Alignement gauche : align-g

La cinquième méthode d'alignement repose sur un modèle un peu plus fin, fondé sur un modèle de Markov caché. Dans ce modèle, les phonèmes sont les états cachés et les graphèmes les états observés. Un état supplémentaire '–' est ajouté de façon à autoriser une insertion dans la chaîne de phonèmes. De même, le symbole '–' est ajouté à l'ensemble des observations possibles. Le HMM considéré prend donc la forme suivante :



Un algorithme de type Viterbi [9] est utilisé pour trouver le meilleur alignement possible. Pour estimer les paramètres du HMM, on part d'un alignement aveugle tel que ceux évoqués ci-dessus. Ces données alignées permettent d'inférer les poids d'un modèle de Markov caché. Ce même modèle permet ensuite de réaligner (plus finement) ces données. Ce procédé est itéré plusieurs fois, jusqu'à la satisfaction d'un certain critère d'arrêt (nombre d'itérations, stabilisation, etc.) ; cette méthode est donc une instanciation du principe EM (espérance-maximisation, Dempster *et al.* (1977)). Sur de telles méthodes d'alignement à partir de HMM, voir par exemple Vogel *et al.* (1996) et Och (2003).

Les différentes stratégies d'alignement envisagées permettent de mettre en évidence la contribution de l'étape d'alignement dans le processus global d'apprentissage.

L'alignement des données effectué, il est possible d'appliquer des méthodes classiques de classification. Les propriétés de cette tâche de classification sont les suivantes : les attributs sont symboliques, le nombre d'attributs est égal à la taille de la fenêtre considérée (de 5 à 11) et il y a à peu près 500 classes [10]. La souplesse de l'algorithme des k-ppv le rend particulièrement adapté à cette tâche ; en particulier, il gère naturellement les problèmes multi-classes et les données symboliques. Le logiciel TIMBL (Daelemans *et al.*, 2004) implémente une version enrichie de cet algorithme [11].

5.2.4 Résultats

Les résultats présentés correspondent aux (micro et macro) rappel, précision et f-score, obtenus sur l'intégralité de la prononciation. Autrement dit, pour que la quantité $f(x_i) \cap y_i$ augmente (de 1), il faut trouver une prononciation exacte ; une lettre bien prononcée ou un morceau de forme bien prononcé ne permettent pas de faire augmenter cette quantité d'une quelconque fraction.

ALANIS

Influence de d Dans cette expérience, nous étudions l'influence du degré maximum autorisé (d) sur les performances. Intuitivement, le choix de ce paramètre correspond à un compromis. En effet, en limitant le degré, on risque de manquer des solutions potentiellement intéressantes ; autoriser un degré élevé conduit à l'introduction d'un certain bruit. Les résultats obtenus sont représentés sur les figures 5.4 et 5.5 : le (macro) rappel et la (macro) précision sont en ordonnée et le degré maxi-

9. L'algorithme canonique est modifié de façon à pouvoir insérer des symboles '–' dans la chaîne en entrée.

10. Ce nombre élevé de 500 classes alors que l'ordre de grandeur du nombre de phonèmes est 50 s'explique de la manière suivante : lors de la phase alignement, un symbole graphique peut être amené à être aligné avec plusieurs phonèmes (présence du symbole '–' dans la chaîne graphique en entrée) ; dans ce cas, chaque couple ou triplet de phonèmes constitue une nouvelle classe. Bien entendu, ces classes sont beaucoup plus rares que celles correspondant à un seul phonème.

11. En particulier, il fournit plusieurs méthodes de pondération des attributs.

mal en abscisse. Un degré maximal de 2, 5 signifie un degré maximal en entrée de 2 et un degré maximal en sortie de 3.

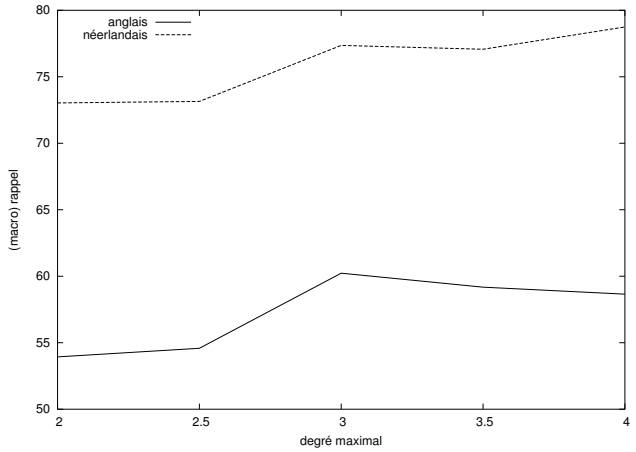


FIG. 5.4 – Influence du paramètre d ($n = 150$) - rappel

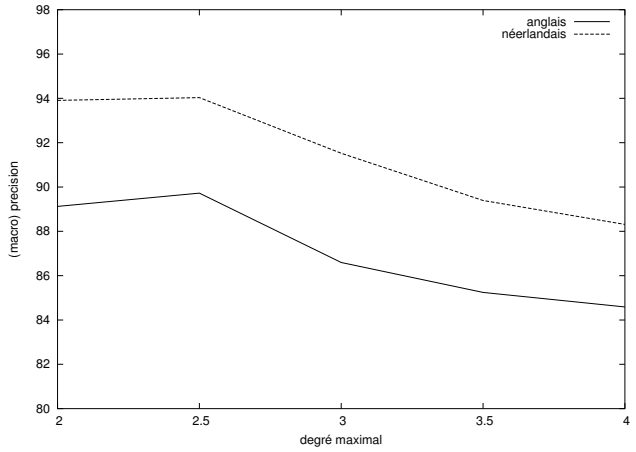


FIG. 5.5 – Influence du paramètre d ($n = 150$) - précision

L'intuition est confirmée par l'expérience puisque le rappel augmente avec d alors que la précision diminue. Autoriser un degré plus élevé conduit à la proposition d'un nombre plus grand de solutions, mais celles-ci peuvent se révéler être de moins bonne qualité. Dans la suite des expériences, sauf lorsque cela sera spécifié, nous fixerons d à 3 en entrée et en sortie.

Influence de n Le nombre de paradigmes tirés aléatoirement pour effectuer l'inférence est le second paramètre du modèle. A priori, plus ce nombre est élevé, plus les performances sont bonnes. Cette intuition est confirmée expérimentalement comme on peut le constater à l'observation des courbes des figures 5.6 et 5.7. Ce paramètre n'est donc pas à optimiser ; il sera déterminé par les besoins de l'application et le temps de calcul disponible. À noter que selon le degré et le nombre de paradigmes tirés, les expériences peuvent demander de 30 minutes à 1 heure pour traiter un millier d'exemples.

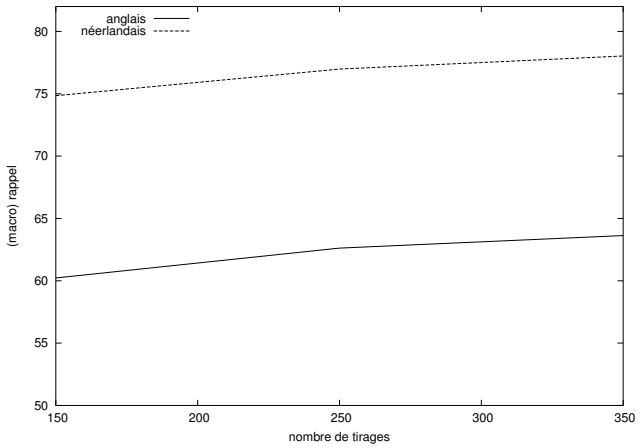


FIG. 5.6 – Influence du paramètre n ($d = 3$) - rappel

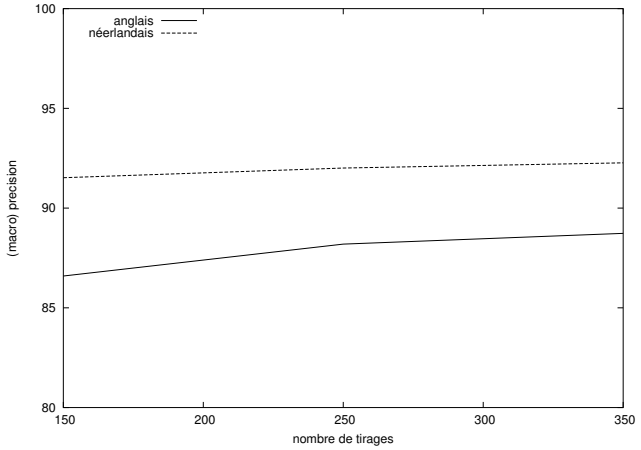


FIG. 5.7 – Influence du paramètre n ($d = 3$) - précision

Comparaison des stratégies Les stratégies de type seuil permettent (i) de jouer sur le compromis rappel/précision, (ii) de mettre en évidence la qualité du classement effectué. Dans cette expérience, le nombre de paradigmes tirés est fixé à 150, et les degrés maximum à 3. Les figures 5.8 et 5.9 illustrent les résultats obtenus en utilisant la stratégie *Threshold*, qui impose un poids minimal pour les solutions. Le rappel et la précision sont en ordonnée et le seuil fixé en abscisse.

La stratégie *KBest* limite le nombre de solutions à k ; les résultats pour cette stratégie s'observent sur les figures 5.10 et 5.11.

Le fait de pouvoir proposer un ensemble de solutions ordonnées est une propriété particulièrement appréciable du modèle, qui permet de privilégier le rappel ou la précision selon les applications traitées. Lorsque l'on dispose de plusieurs solutions classées, ce contrôle rappel/précision est souvent rencontré. Toutefois, signalons que ces courbes disent plus qu'elles n'y laissent paraître : le fait d'avoir ce contrôle rappel/précision indique *justement* que le classement effectué est de

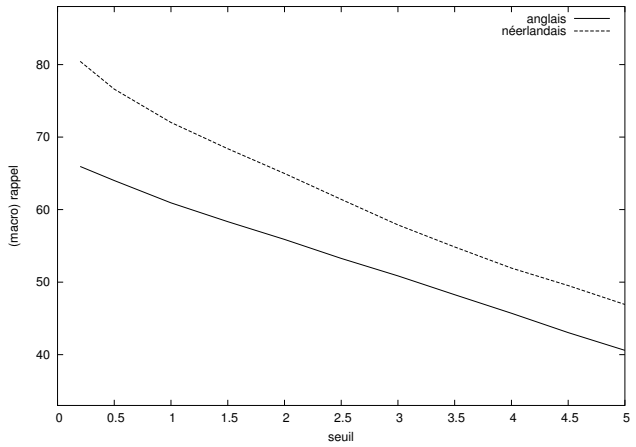


FIG. 5.8 – Influence du seuil - *Threshold* - rappel

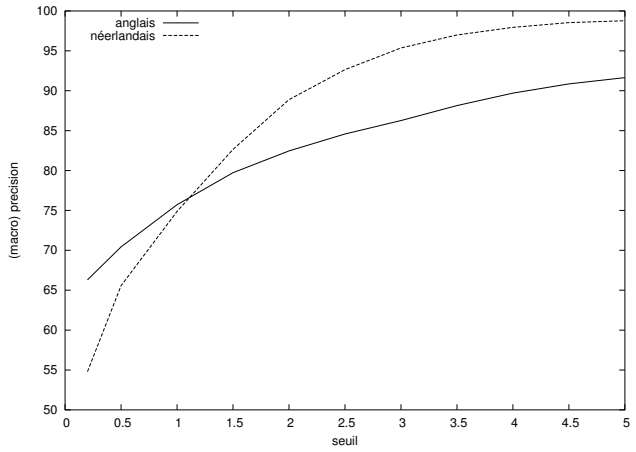


FIG. 5.9 – Influence du seuil - *Threshold* - précision

bonne qualité. Les meilleures solutions sont bien en tête de classement, ce qui valide le critère utilisé pour pondérer les solutions. Les stratégies *AE* et *Mean* déterminent automatiquement le nombre de solutions à proposer ; sur de nombreuses tâches, elles se rapprochent de la stratégie optimale *Monkey*. C'est cette dernière qui sera utilisée dans la suite, car comme nous l'avons déjà signalé, elle représente le potentiel de la méthode APPA. En outre, lorsque l'ambiguïté est faible, fixer le nombre maximal de solution à 1 suffit.

TIMBL

Influence de l'alignement Quatre méthodes (aveugles) d'alignements ont été proposées pour reformuler le problème de la prononciation automatique en une tâche de classification (gauche, centre, droit et aléatoire). Chacune de ces méthodes peut servir de tremplin à l'estimation des paramètres d'un HMM lui-même en mesure d'aligner les données. Les résultats (macro rappel et précision) obtenus

cf. Sec. 5.2.3, p. 129

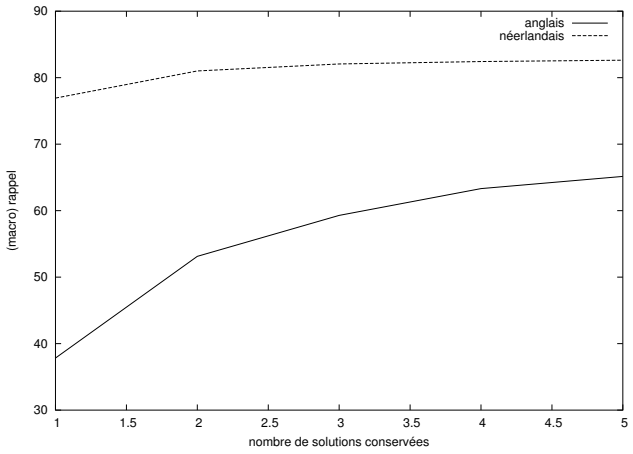


FIG. 5.10 – Influence du seuil - *KBest* - rappel

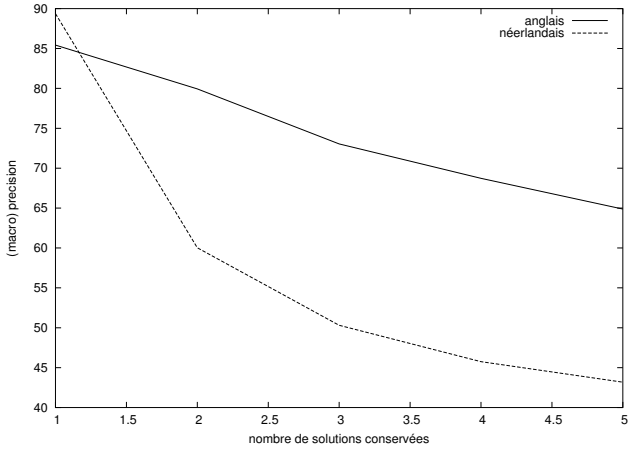


FIG. 5.11 – Influence du seuil - *KBest* - précision

sont reportés dans le tableau 5.2, dans lequel SH signifie *sans HMM* (alignement aveugle simple) et AH *avec HMM* (alignement aveugle servant de tremplin à un HMM).

	gauche		centre		droit		aléatoire	
	SH	AH	SH	AH	SH	AH	SH	AH
Macro rappel	12,94	45,95	21,30	45,39	17,17	44,09	10,42	47,40
Macro précision	17,97	70,53	31,59	70,19	24,34	66,30	15,79	73,83

TAB. 5.2 – Influence de l'alignement (anglais)

L'apprentissage effectué suite à un alignement aveugle conduit à des résultats très nettement moins bons que ceux obtenus après un alignement fondé sur un HMM : l'apport de l'alignement par HMM varie de +113% à +355% pour le rappel et de +122% à +367% pour la précision. Ces résultats mettent clairement en évidence le poids de l'alignement dans le processus d'apprentissage. Quand

celui-ci est mauvais, l'apprentissage est quasiment impossible ; par conséquent, TIMBL n'apprend pas tout seul. Nous avions déjà évoqué ce problème lorsque nous avions décrit cette méthode. L'alignement effectué par le HMM correspond bien à un « ajout d'information » (à l'aide d'une induction) par rapport aux données initiales, et la reformulation impliquée cache en réalité une partie de la réponse apportée au problème. Preuve supplémentaire : on peut utiliser directement le HMM induit lors de la phase d'alignement pour prononcer ; en effet, trouver la séquence d'états (les phonèmes) la plus probable étant donnée une séquence d'observations (les graphèmes) s'obtient directement par l'algorithme de Viterbi. Les résultats obtenus de cette façon sont assez mauvais ($\simeq$ 5% de F-score), mais largement meilleurs que le hasard, trace d'un début d'apprentissage. En outre, alors que TIMBL est capable de stocker des centaines de milliers d'instances et de traiter des milliers d'instance en quelques minutes, l'alignement de chaque jeu de données à l'aide d'un HMM requiert un temps d'exécution de l'ordre de l'heure, élément qui appuie la non-trivialité de la tâche. Nous retiendrons finalement que le système TIMBL est souvent utilisé conjointement à un autre mécanisme et que c'est la combinaison de ces mécanismes qui fournit les résultats obtenus, et non TIMBL seul [12].

cf. Sec. 3.1.2, p. 45

Comparaison de ALANIS et de TIMBL+HMM Les résultats globaux respectifs obtenus par les systèmes ALANIS et TIMBL sur la tâche de prononciation sont présentés dans les tableaux 5.3 et 5.4 [13].

	micro rappel	macro rappel	micro précision	macro précision
ALANIS	49,66	63,63	62,87	88,73
TIMBL	60,10	47,40	65,70	73,83

TAB. 5.3 – Tâche de prononciation - ALANIS vs. TIMBL - anglais

	micro rappel	macro rappel	micro précision	macro précision
ALANIS	78,03	80,55	90,10	92,27
TIMBL	93,87	93,84	93,88	93,89

TAB. 5.4 – Tâche de prononciation - ALANIS vs. TIMBL - néerlandais

Pour le néerlandais, les résultats obtenus sont relativement bons, mais inférieur toutefois à ceux issus de TIMBL, notamment au niveau du rappel. En ce qui concerne l'anglais, en raison de la présence d'ambiguïtés, une grande différence est

12. Signalons par ailleurs qu'il nous semble contradictoire de défendre prioritairement une approche à base de mémoire d'une part, et de devoir faire appel à des méthodes inductives d'autre part.

13. Pour ALANIS, le nombre de paradigmes tirés est 350 et le degré maximal autorisé 3 ; pour TIMBL, une batterie de tests avec différents paramètres a été lancée et les résultats présentés correspondent aux résultats obtenus avec la meilleure combinaison de paramètres.

observable entre les micro et les macro-mesures. Sur les premières, TIMBL présente de meilleurs performances qu'ALANIS ; c'est l'inverse pour les macro-mesures. Sur les macro-mesures, TIMBL est pénalisé par son manque de souplesse car il ne peut proposer qu'une solution. Cette propriété s'observe plus particulièrement sur les entrées ambiguës, pour lesquelles la tâche est encore plus difficile (cf. tableau 5.5).

	micro rappel	macro rappel	micro précision	macro précision
ALANIS	26,18	61,32	36,37	90,08
TIMBL	30,41	29,48	42,64	68,45

TAB. 5.5 – Tâche de prononciation - ALANIS vs. TIMBL - anglais - entrées ambiguës

Pour comprendre plus précisément le comportement des algorithmes, il est utile de ventiler les résultats selon la partie du discours (cf. tableaux 5.6 et 5.7) : en ce qui concerne ALANIS, on observe clairement une différence entre les entrées appartenant aux catégories Verbes, Noms et Adjectifs d'une part et celles n'y appartenant pas d'autre part. Ces résultats sont en cohérence avec notre approche,

		micro rappel	macro rappel	micro précision	macro précision
Noms	ALANIS	51,41	70,69	62,17	90,36
	TIMBL	59,69	45,29	64,91	70,25
Verbes	ALANIS	57,53	82,64	59,50	84,75
	TIMBL	67,20	58,12	72,70	85,07
Adjectifs	ALANIS	26,46	43,80	58,17	95,90
	TIMBL	57,20	44,04	62,83	71,49
Autres	ALANIS	5,62	19,22	55,76	96,45
	TIMBL	42,08	35,69	49,56	64,54

TAB. 5.6 – Tâche de prononciation - ALANIS vs. TIMBL - anglais
ventilation selon les parties du discours

		micro rappel	macro rappel	micro précision	macro précision
Noms	ALANIS	66,50	70,60	87,59	90,67
	TIMBL	91,89	91,85	91,90	91,91
Verbes	ALANIS	87,77	92,10	89,17	93,43
	TIMBL	95,81	95,80	95,84	95,87
Adjectifs	ALANIS	88,72	92,59	92,42	96,67
	TIMBL	95,54	95,42	95,55	95,54
Autres	ALANIS	17,31	40,22	67,30	80,62
	TIMBL	96,88	96,88	96,88	96,88

TAB. 5.7 – Tâche de prononciation - ALANIS vs. TIMBL - néerlandais
ventilation selon les parties du discours

qui s'appuie fortement sur l'existence de relations paradigmatiques entre les entités linguistiques. Dans cette tâche, les entrées sont regroupées en familles lexématiques ; lorsque les paradigmes flexionnels sont pauvres ou inexistants, l'apprentissage n'a pas lieu. L'étude de l'influence du regroupement effectué constitue un travail en cours.

5.3 Analyse flexionnelle

5.3.1 Présentation de la tâche

Dans cette tâche, il s'agit d'effectuer une analyse flexionnelle d'une forme graphique, c'est-à-dire lui associer un lexème ainsi qu'un certain nombre de traits morpho-syntaxiques : catégorie grammaticale, genre, nombre, cas, temps, etc. Un exemple de paire d'entrée/sortie pour cette tâche est :

$$\text{entrée=}\textit{marcherons}\text{; sortie=} \begin{bmatrix} \textit{Lexème} & : \textit{marcher} \\ \textit{Catégorie} & : \textit{verbe} \\ \textit{Personne} & : 1^{\text{re}} \\ \textit{Nombre} & : \textit{pluriel} \\ \textit{Temps} & : \textit{futur de l'indicatif} \end{bmatrix}.$$

Les traits morpho-syntaxiques étudiés sont en nombre fini, et pour chacun des traits, l'ensemble des valeurs possibles est également fini. Un ensemble de traits peut donc être représenté par un vecteur d'attributs symboliques, ce qui donne pour la même paire d'entrée/sortie :

$$\text{entrée=}\textit{marcherons}\text{; sortie=\{}\textit{marcher}\text{, }\texttt{V1pf-}\text{\}},$$

où la première composante du vecteur désigne la partie du discours (`V`=verbe), la deuxième la personne (`1`=1re personne), la troisième le nombre (`p`=pluriel), la quatrième le temps (`f`=futur de l'indicatif), et la cinquième le genre (`-`=non pertinent).

Ce genre d'analyse présente un intérêt pour de nombreuses applications. En particulier, il est utile, pour un étiqueteur morpho-syntaxique, de savoir déterminer l'ensemble des parties du discours possibles d'un mot inconnu (Mikheev, 1997).

5.3.2 Données

Les données utilisées pour cette tâche sont également issues du corpus CELEX ; leurs propriétés sont détaillées dans le tableau 5.8.

5.3.3 Formalisation et pré-traitements

Formalisation nécessaire à APPA Dans cette tâche, l'entrée est représentée par une séquence de symboles graphiques : on peut donc choisir comme structure algébrique sous-jacente la structure de monoïde libre sur l'alphabet des symboles

Corpus	Langue	Nombre d'entrées	Nombre d'entrées ambiguës	Pourcentage d'entrées ambiguës	Nombre moyen d'analyses
CELEX	Anglais	89412	17320	19,37%	1,80
	Néerlandais	324249	39332	12,13%	1,72
	Allemand	342452	8897	2,60%	1,05

TAB. 5.8 – Propriétés des données, tâche d'analyse flexionnelle

graphiques. La sortie est constituée d'un ensemble de traits : la structure algébrique adoptée sera l'ensemble des structures de traits (plates). Cette application permet, elle aussi, de mettre en évidence la souplesse du modèle. En effet, aucune reformulation particulière des données n'est requise ; le traitement de tâches différentes s'effectue à partir d'un même moteur d'inférence, qui nécessite uniquement que l'on se place dans un certain contexte algébrique. Les entrées sont également regroupées en familles flexionnelles.

Pré-traitement nécessaire à TIMBL Dans cette tâche, il est difficile d'effectuer un alignement entre l'entrée et la sortie de manière à se ramener à plusieurs problèmes de classification ; la tâche de classification doit donc s'effectuer « en bloc », c'est-à-dire sur l'intégralité de la forme. Puisque la méthode d'apprentissage attend des vecteurs d'attributs de taille fixe en entrée, les formes graphiques doivent être tronquées ou complétées de manière à obtenir des vecteurs d'attributs symboliques de taille fixe. Nous envisageons trois possibilités : les formes sont alignées sur la gauche, au centre, ou sur la droite (cf. figure 5.12). Cette étape de reformu-

forme 1	m	a	r	c	h	e	r	o	n	s
forme 2	-	p	a	r	t	i	r	o	n	s
forme 3	e	c	e	v	r	a	i	e	n	t
etc.										

FIG. 5.12 – Alignement droit - taille fixe de 10

lation illustre une nouvelle fois le côté peu naturel de l'approche : aucun de ces alignements ne s'impose dans l'absolu. Toutefois, étant donné le fonctionnement de la flexion dans les langues indo-européennes, qui repose principalement sur des suffixations, l'alignement « sur la droite » semble plus prometteur.

À chacun de ces vecteurs symboliques est associée une classe, qui, ici encore, représente l'information de sortie « en bloc ». Cette information contient une signature permettant de passer de la forme au lexème, et l'ensemble des traits morpho-syntaxiques. Chaque classe est considérée singulièrement ; en particulier, il n'est pas possible d'exploiter le fait que deux classes sont en relation.

5.3.4 Résultats

TIMBL

Influence de l'alignement De façon prévisible, le choix de l'alignement est particulièrement déterminant dans la tâche d'apprentissage : l'alignement droit est meilleur que l'alignement centre, lui-même meilleur que l'alignement gauche (cf. tableau 5.9).

	F-score
Alignement droit	76,78
Alignement centre	50,84
Alignement gauche	38,86

TAB. 5.9 – Tâche d'analyse flexionnelle - anglais

Comparaison de ALANIS et de TIMBL

Nous avons effectué un certain nombre d'expériences en relation avec cette tâche. En particulier, nous avons également étudié l'influence des paramètres d et n. Nous ne reproduisons pas ici les courbes mettant en évidence cette influence ; elles conduisent aux conclusions précédemment évoquées, à savoir : le rappel augmente avec le degré maximal alors que la précision diminue et ils augmentent tous les deux avec le nombre de paradigmes tirés aléatoirement. Dans les résultats présentés, le paramètre d est fixé à 3, et n à 150 ; sur ces résultats, voir également Stroppa & Yvon (2005),

cf. Sec. 5.2.4, p. 130

Les résultats obtenus sont illustrés par les tableaux 5.10, 5.11, et 5.12.

		micro rappel	macro rappel	micro précision	macro précision
Noms	ALANIS	75,26	76,33	95,37	85,19
	TIMBL	76,06	65,29	79,67	91,34
Verbes	ALANIS	94,79	93,46	97,37	94,49
	TIMBL	33,78	25,65	42,36	76,99
Adjectifs	ALANIS	27,89	36,91	87,67	71,15
	TIMBL	57,61	48,98	62,86	81,64

TAB. 5.10 – Tâche d'analyse flexionnelle - anglais

Ces résultats montrent que la tâche, du point de vue de la classification, n'est pas triviale. De très bons scores peuvent être atteints par TIMBL (adjectifs allemands, noms néerlandais), mais sont très dépendants de la langue et de la catégorie. Par exemple, alors que les adjectifs allemands sont plutôt reconnus correctement (89% de rappel et de précision), les verbes le sont beaucoup moins (57% de rappel et de précision). En outre, il existe une certaine homogénéité entre le rappel et la précision. Cela est cohérent avec la non-gestion de l'ambiguïté : puisqu'une

		micro rappel	macro rappel	micro précision	macro précision
Noms	ALANIS	54,59	55,25	74,75	67,77
	TIMBL	85,39	82,65	86,24	88,50
Verbes	ALANIS	93,26	93,59	94,36	91,20
	TIMBL	45,82	43,97	50,79	61,74
Adjectifs	ALANIS	90,02	89,16	95,33	86,83
	TIMBL	76,75	73,69	78,29	81,22

TAB. 5.11 – Tâche d'analyse flexionnelle - néerlandais

		micro rappel	macro rappel	micro précision	macro précision
Noms	ALANIS	77,32	73,30	81,70	78,17
	TIMBL	80,95	80,06	81,45	82,01
Verbes	ALANIS	90,50	88,62	90,63	87,78
	TIMBL	56,49	55,40	57,21	58,33
Adjectifs	ALANIS	99,01	98,90	99,15	89,31
	TIMBL	89,31	88,71	89,57	89,84

TAB. 5.12 – Tâche d'analyse flexionnelle - allemand

et une seule solution est proposée par entrée, une entrée ambiguë conduit à une diminution à la fois de la précision et du rappel.

En ce qui concerne ALANIS, on remarque, ici encore, qu'il s'appuie essentiellement sur des paradigmes riches ; en particulier, la pauvreté de la flexion des adjectifs anglais entraîne un faible taux de rappel. En outre, on observe un comportement assez conservateur ; en effet, quel que soit le rappel, la précision est toujours assez élevée (plus mauvais score de 67% pour les noms en néerlandais), signifiant que le système préfère se taire à se tromper. À l'inverse, TIMBL propose toujours une solution, et sa précision peut descendre jusqu'à 42% (verbes anglais).

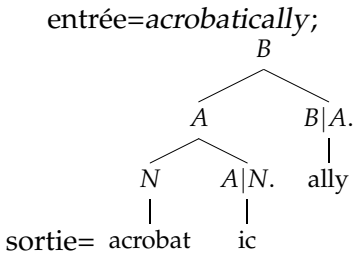
Ces résultats illustrent à la fois la souplesse du modèle (gestion naturelle d'objets structurés, d'entrées ambiguës) et son efficacité (capacité à traiter des problèmes non triviaux pour la classification).

5.4 Analyse dérivationnelle

5.4.1 Présentation de la tâche

L'objectif de cette tâche est d'effectuer une analyse dérivationnelle d'un lexème représenté par une forme graphique canonique. Cette analyse correspond à une structure hiérarchique correspondant à la trace du processus du dérivation du

lexème. Un exemple de paire d'entrée/sortie pour cette tâche est :



Dans la structure hiérarchique de sortie, les feuilles constituent les morphèmes composant le lexème. Les nœuds sont étiquetés par des catégories grammaticales. Un morphème étiqueté par une catégorie de la forme $A|N.$ est un morphème de type « affixe » ; une catégorie telle que $A|N.$ signifie que le morphème en question est suffixé (.) à un nom (N) pour former un adjectif (A). Disposer des analyses dérivationnelles des formes que l'on rencontre fournit une information exploitable dans plusieurs contextes : aide à la recherche d'information (lemmatisation, racinisation), aide à la compréhension, aide à la traduction, etc.

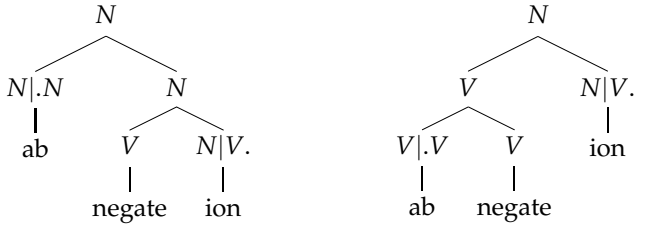
5.4.2 Données

Les données utilisées pour cette tâche sont issues du lexique CELEX ; leurs propriétés sont détaillées dans le tableau 5.13.

Corpus	Langue	Nombre d'entrées	Nombre d'entrées ambiguës	Pourcentage d'entrées ambiguës	Nombre moyen d'analyses
CELEX	Anglais	46129	7309	15,84%	1,18
	Néerlandais	120967	8651	7,15%	1,06
	Allemand	49936	1415	2,83%	1,02

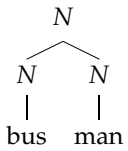
TAB. 5.13 – Propriétés des données, tâche d'analyse dérivationnelle

Exemple d'entrée ambiguë L'entrée *abnegation* est associée à deux analyses possibles, représentées par les deux structures hiérarchiques suivantes :

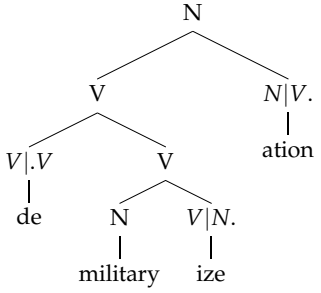


5.4.3 Formalisation et pré-traitements

Formalisation nécessaire à APPA Dans cette tâche, l'entrée est encore une séquence de symboles graphiques : on peut donc choisir comme structure algébrique sous-jacente la structure de monoïde libre sur l'alphabet des symboles graphiques. La sortie est une structure hiérarchique : la structure algébrique adoptée sera l'ensemble des arbres. L'algorithme utilisé pour résoudre des équations analogiques entre arbres est l'algorithme approximatif présenté dans la section 4.2.5. Dans cette tâche, les entrées sont regroupées par « famille dérivationnelle » : chacune de ces familles est représentée par une « racine ». Nous avons considéré qu'une racine est une feuille dont la catégorie n'est pas une catégorie de type affixe [14]. Une entrée appartient à toutes les familles correspondant aux racines qui la composent. Par exemple, *acrobatically* appartient à la famille *acrobat* ; *busman* appartient aux familles *bus* et *man*.

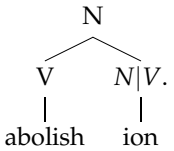


Pré-traitement nécessaire à TIMBL Pour pouvoir résoudre cette tâche à l'aide de TIMBL, il faut, ici encore, opérer une reformulation des données. La méthode employée (i) représente les arbres sous une certaine forme linéarisée, (ii) applique un alignement entre les séquences en entrée et en sortie. Dans cette tâche, l'alignement impliqué est légèrement différent de celui introduit dans la tâche de prononciation ; en effet, alors que l'alphabet des symboles graphiques et celui des phonèmes sont deux alphabets distincts, les feuilles des structures hiérarchiques sont des séquences de symboles appartenant à l'alphabet sous-jacent aux séquences en entrée. En outre, il arrive fréquemment que la séquence en entrée corresponde à la concaténation des feuilles de la structure hiérarchique associée en sortie (on parlera de *frontière*) : c'est le cas des deux exemples évoqués plus haut (*acrobatically=acrobat+ic+ally* et *busman=bus+man*). Il semble donc raisonnable de chercher à mettre en relation la séquence en entrée avec les feuilles. Cet alignement est plus difficile lorsque le morphème impliqué dans la décomposition ne se retrouve pas identiquement dans la forme ; dans ce cas, on parle d'allomorphie. Par exemple, la décomposition associée à la forme *demilitarization* est la suivante :



14. Le critère réel est un peu plus complexe, mais cela n'est pas déterminant pour la compréhension de l'approche.

Dans ce cas, le morphème *military* apparaît sous la forme *militar*, et le morphème *ize* sous la forme *iz* ; la frontière n'est pas exactement égale à la forme impliquée (*militarization≠military+ize+ation*). Le même phénomène intervient pour *abolition* :



De façon à représenter séquentiellement les structures, deux types de linéarisations sont considérés : les expressions parenthésées, et les linéarisations fondées sur l'arité, toutes deux introduites dans la section 4.2.5. Par exemple, l'expression parenthésée associée à la structure hiérarchique correspondant à *abolition* est :

cf. Sec. 4.2.5, p. 112

$$(N(V(abolish))(N|V.(ion)))$$

L'alignement fondé sur l'arité prend une forme telle que :

$$2N + 1V + 0a, b, o, l, i, \_s, h + 1N|V. + 0i, o, n$$

Dans les deux cas, on commence par aligner la séquence en entrée et la frontière de la structure, à l'aide d'un algorithme d'alignement classique reposant sur la distance d'édition (Wagner & Fischer, 1974). Par exemple, pour *abolition*, on peut avoir l'alignement suivant :

a	b	o	l	i	t	-	i	o	n
a	b	o	l	i	s	h	i	o	n

Ensuite, les morceaux de la chaîne correspondant au « squelette » de la structure sont ajoutés aux endroits adéquats, ce qui donne :

-	-	-	-	-	a	b	o	l	i	t	-	-	-	-	-	-	i	o	n	-	-
(	N	(	V	(	a	b	o	l	i	s	h	)	)	(	N|V.	(	i	o	n	)	)	)

Une fois l'alignement effectué, on utilise la méthode de fenêtrage précédemment décrite ; la méthode de classification peut ensuite s'appliquer. Notons que les cas d'allomorphie permettent de mettre en évidence une nouvelle fois la difficulté de faire correspondre des unités de différents espaces de représentation. Par ailleurs, il est important de noter que la sortie proposée par le système peut ne pas correspondre à une linéarisation d'arbre. En effet, puisque chaque fenêtre est considérée de manière indépendante, rien n'empêche le système de proposer la sortie $((ion($; dans ce cas, on considère que le système ne propose pas de sortie (pas de pénalisation inutile sur la précision).

cf. Sec. 3.1.2, p. 45

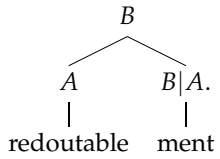
5.4.4 Résultats

Influence de l'alignement La présence de nombreuses parenthèses rend la tâche de classification particulièrement difficile dans le cas des expressions parenthésées ; en effet, les parenthèses indiquent des dépendances à longue distance qui

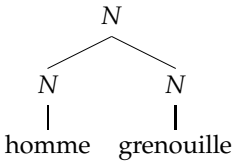
perturbent la tâche de classification agissant localement (dû au fenêtrage). Le rappel obtenu par TIMBL avec cette linéarisation est proche de zéro. Cela signifie que toutes les sorties qu'il propose ne sont pas des linéarisations d'arbres valides (mauvaise correspondance des parenthèses). Dans le cas de la linéarisation fondée sur l'arité, il n'y a plus de dépendance à long terme et les résultats sont beaucoup plus acceptables (ce sont ceux présentés dans la suite). On observe, ici encore, que l'influence du choix d'une certaine reformulation est réellement déterminant.

Comparaison de ALANIS et de TIMBL

Les tableaux 5.14, 5.15 et 5.16 représentent les résultats obtenus respectivement par ALANIS et TIMBL sur cette tâche. Les entrées sont distinguées selon qu'elles font intervenir de l'affixation et/ou de la composition. Une affixation correspond à l'ajout d'une feuille de type affixe, comme dans



La composition accole deux structures a priori autonomes :



Dans cette tâche, nous cherchons à apprendre des structures entières ; des sous-arbres corrects ne suffisent pas à faire augmenter le rappel et la précision ; cette tâche est donc particulièrement difficile.

		micro rappel	macro rappel	micro précision	macro précision
Composition	ALANIS	19,64	19,53	88,33	91,49
	TIMBL	34,66	33,84	59,43	61,52
Affixation	ALANIS	56,09	56,67	81,21	84,05
	TIMBL	6,48	6,17	59,65	64,09
Composition+Affixation	ALANIS	17,21	16,26	84,68	89,99
	TIMBL	10,99	9,86	74,86	82,24
Autre	ALANIS	14,60	17,36	67,56	74,90
	TIMBL	11,36	11,05	64,41	72,73

TAB. 5.14 – Tâche d'analyse dérivationnelle - anglais

Un élément observable est la différence de précision entre les deux systèmes. En effet, la précision générale d'ALANIS est relativement élevée étant donnée la

		micro rappel	macro rappel	micro précision	macro précision
Composition	ALANIS	39,02	39,54	88,58	90,39
	TIMBL	44,97	45,04	77,93	80,39
Affixation	ALANIS	54,31	55,82	88,88	89,56
	TIMBL	8,80	8,84	73,94	76,15
Composition+Affixation	ALANIS	30,73	31,78	84,64	88,13
	TIMBL	25,47	23,46	80,14	84,55
Autre	ALANIS	35,35	40,04	72,57	78,32
	TIMBL	66,30	65,20	87,06	88,24

TAB. 5.15 – Tâche d'analyse dérivationnelle - néerlandais

		micro rappel	macro rappel	micro précision	macro précision
Composition	ALANIS	19,28	20,56	94,11	95,16
	TIMBL	24,95	24,73	68,53	69,23
Affixation	ALANIS	36,55	36,82	88,25	88,59
	TIMBL	12,95	13,02	71,91	72,20
Composition+Affixation	ALANIS	8,41	8,53	92,24	95,60
	TIMBL	12,95	13,02	71,91	72,21
Autre	ALANIS	17,97	18,33	80,45	81,27
	TIMBL	55,59	55,17	81,69	82,02

TAB. 5.16 – Tâche d'analyse dérivationnelle - allemand

difficulté de la tâche. Elle est pratiquement systématiquement plus élevée que celle atteinte par TIMBL. Cela s'opère parfois au détriment du rappel ; on retrouve ici le caractère conservateur du système, qui ne se prononce qu'avec une certaine assurance. En outre, il essaie, ici encore, de s'appuyer sur des paradigmes riches ; l'exploitation des proportions est plus aisée dans les cas d'affixation, qui font apparaître des proportions telles que

véritable : *véritablement* :: *redoutable* : *redoutablement*.

5.5 Conclusion

La méthode d'apprentissage APPA a été implantée dans le logiciel ALANIS, que nous avons développé à cet effet. Pour rendre cette méthode effective, nous avons proposé des solutions aux problèmes de la recherche de triplets et du classement des solutions.

Ce système a été testé sur plusieurs tâches de TAL : la prononciation, l'analyse flexionnelle et l'analyse dérivationnelle. Ces trois tâches impliquent des objets

structurés en entrée et en sortie. Nous avons comparé notre approche à un système de classification de l'état de l'art (TIMBL) ; en particulier, nous montrons les problèmes que peut poser et l'influence que peut avoir la tâche de reformulation nécessaire à l'utilisation des méthodes de classification. Plus les objets sont structurés, plus cette reformulation est difficile et peu naturelle. À l'inverse, le système ALANIS peut gérer sans adaptation des objets structurés, à la fois en entrée et en sortie. Cette propriété le rend particulièrement souple d'utilisation. Par ailleurs, les ambiguïtés non sont pas maltraitées, mais considérées pleinement puisqu'il lui est possible de proposer un ensemble de solutions (pondérées). Cet ensemble peut être vide, ce qui signifie que le système ne se prononce pas : la précision est alors privilégiée par rapport au rappel. Outre la souplesse du système, les expériences effectuées permettent de mettre en évidence son efficacité. Les résultats obtenus sont prometteurs, compte tenu de la difficulté des tâches et de la simplicité actuelle du moteur d'inférence.

CHAPITRE

6 Conclusion et perspectives

Dans le travail effectué dans le cadre de cette thèse, nous avons défini et caractérisé des modèles à base d'analogies pour effectuer un Apprentissage Automatique d'un certain nombre de tâches liées au Traitement Automatique des Langues. En particulier, notre attention a porté spécifiquement sur celles s'exprimant naturellement comme des changements de niveau de représentation. Nous avons défendu l'idée qu'une exploitation de l'organisation paradigmatique des données linguistiques, en particulier à l'aide d'extraction de proportions analogiques, permet d'effectuer une inférence pouvant être à la fois souple et efficace.

Nous avons tout d'abord resitué la notion d'apprentissage par analogie dans différents contextes. Les différents points de vue adoptés ont offert un éclairage sur les éléments constitutifs de cette notion. En particulier, nous avons étudié le cas du raisonnement par analogie dans le domaine des Sciences Cognitives et discuté des liens que pouvait entretenir un tel type de raisonnement avec les méthodes d'apprentissage dites paresseuses issues de l'Apprentissage Automatique. Enfin, nous avons succintement évoqué la notion de proportion analogique dans un contexte linguistique, ainsi que certaines approches à partir d'exemples en TAL. De manière générale, l'approche analogique repose sur une inférence s'effectuant directement à l'aide d'exemples déjà analysés, et ne nécessite pas de généralisation ou d'abstraction. Elle s'oppose en cela aux approches déductives et inductives plus communes.

Les tâches de TAL qui ont plus particulièrement retenu notre attention, à savoir celles s'exprimant comme un changement de niveau de représentation, correspondent à des problèmes difficiles pour qui veut les voir comme des problèmes d'Apprentissage Automatique. En particulier, elles nécessitent de savoir manipuler naturellement des objets structurés, elles doivent présenter une certaine forme de symétrie entre l'entrée et la sortie et on peut attendre d'elles qu'elles proposent des solutions qu'elles n'ont encore jamais rencontrées. Une méthode d'apprentissage exploitant des proportions analogiques a été étudiée et nous avons montré comment celle-ci permet de répondre à ces questions spécifiques. Par ailleurs, nous avons introduit la notion d'extension analogique, qui permet d'exprimer simplement le biais d'apprentissage de la méthode.

Ensuite, nous avons étudié la notion de proportion analogique d'un point de vue formel : quels sont les éléments permettant de déterminer si quatre termes forment une proportion ? Pour répondre à cette question, nous sommes partis d'un modèle de proportions entre chaînes, éléments d'un monoïde libre, que nous avons étendu de façon à ce qu'il puisse traiter d'autres types d'objets structurés. Le modèle de proportion proposé s'appuie sur des factorisations et des alternances deux-à-deux de facteurs ; il fournit un cadre général s'appliquant aux semigroupes, aux magmas, ainsi qu'à leurs dérivés. En instanciant ce modèle général, il est possible d'obtenir une définition applicable à une large gamme de structures algébriques. Cela nous permet de traiter le cas des représentations courantes d'entités linguistiques, à savoir les chaînes, les langages et les structures de traits. Le modèle proposé présente une cohérence globale relativement forte et fournit un socle définitionnel solide ; en revanche, il ne fournit pas, dans le cas général, de solution explicite aux problèmes computationnels sous-jacents à l'identification de proportions. Nous avons pu établir un certain nombre de résultats qui répondent à ce problème spécifique dans certains cas, notamment ceux des structures de traits et des arbres. De nombreuses voies restent à explorer pour dire davantage sur ce modèle.

Enfin, nous avons validé expérimentalement l'approche adoptée sur plusieurs tâches de TAL qui s'expriment comme des changements de niveau de représentation : prononciation, analyse flexionnelle et analyse dérivationnelle. Les résultats obtenus sont comparés à un classifieur de l'état de l'art. Ils sont encourageants et les expériences effectuées permettent de mettre en évidence la souplesse du système, en mesure de traiter directement les objets structurés et les ambiguïtés. En comparaison, l'approche fondée sur la classification demande de lourds prétraitements et s'adapte difficilement au traitement d'objets complexes.

Perspectives

Ce travail de recherche laisse envisager de nombreuses perspectives de continuation.

La première extension à court terme concerne les validations expérimentales. Les premiers résultats obtenus sont encourageants, mais méritent d'être complétés, principalement selon trois directions. Tout d'abord, à partir des tâches envisagées, il serait utile de qualifier plus précisément le comportement de l'algorithme d'apprentissage, en mesurant par exemple l'évolution de ses performances en fonction du nombre d'exemples présentés. Ensuite, puisque la méthode proposée est relativement indépendante de la langue, il s'agirait de l'appliquer à un ensemble de langues plus large. Enfin, d'autres tâches sont à étudier : l'analyse syntaxique, la traduction automatique, l'aide à la constitution de ressources pour des langues peu dotées, etc.

Nous avons proposé des solutions pour permettre l'application effective de la méthode APPA, mais tout un ensemble de pistes différentes ou complémentaires sont à envisager de façon à améliorer ses performances, aussi bien en justesse de prédiction qu'en temps d'exécution. Par exemple, au niveau de la recherche de tri-

plets, d'autres types de regroupement peuvent être envisagés. Concernant le tirage aléatoire des paradigmes, il serait intéressant de connaître l'influence d'un tirage non uniforme : est-il profitable de favoriser les paradigmes entrant fréquemment en jeu dans des proportions car c'est eux qui aident à la constitution des proportions ? de les défavoriser de façon à ce qu'ils ne concentrent pas toute l'attention ? Les mêmes types de questions se posent concernant les exemples de la base : y a-t-il un apport à exploiter des informations fréquentielles disponibles pour pondérer a priori les instances ? à utiliser l'implication des instances dans les proportions pour les pondérer ?

Enfin, la modélisation formelle proposée pour rendre compte des proportions analogiques fournit un socle qui n'est qu'un point de départ. À partir de celui-ci, il est possible de construire des proportions potentiellement complexes (proportions entre séquences d'ensembles, séquences d'arbres, etc.). Il reste à caractériser plus précisément les cas pour lesquels ces modèles donnent lieu à des algorithmes efficaces, approximatifs ou exacts ; la question principale étant : quelles sont les hypothèses qui permettent de réduire la complexité des calculs impliqués ?

ANNEXE

A

Les proportions vues comme des similarités de relations

La modélisation des proportions analogiques a fait l'objet d'un certain nombre d'études dans le domaine du traitement automatique du langage naturel, pour des applications telles que l'identification de relations sémantiques entre un nom et un modifieur (Turney *et al.*, 2003 ; Turney & Littman, 2005), l'acquisition automatique de liens morphologiques (Hathout, 2001 ; Claveau & L'Homme, 2005), la prononciation automatique (Yvon, 1996, 1997, 1999), l'analyse morphologique (Stroppa & Yvon, 2005), l'analyse syntaxique (Lepage, 1999) ou la traduction automatique (Lepage & Denoual, 2005). Dans cette annexe, nous décrivons quelques-uns de ces modèles.

Le modèle de Turney & Littman De manière à définir la proportion analogique dans un contexte sémantique, Turney & Littman s'appuient sur la distinction entre similarité de relations et similarité d'attributs : deux paires de mots sont *analogues* si elles impliquent des relations semblables ; deux mots sont *synonymes* s'ils partagent beaucoup d'attributs.

cf. Sec. 2.1.2, p. 18

> *Relational similarity* is correspondence between relations, in contrast with *attributional similarity*, which is correspondence between attributes [...]. When two words have a high degree of attributional similarity, we say they are *synonymous*. When two pairs of words have a high degree of relational similarity, we say they are *analogous*.
>
> Turney & Littman (2005)

Dans ce cadre, une proportion $x : y :: z : t$ est d'autant plus valide que la relation liant x et y est « proche » de celle liant z et t. Turney illustre cette notion à l'aide de la proportion sémantique suivante : « *le maçon* est à la *pierre* ce que *bûcheron* est au *bois* ».

En notant $r_{x,y}$ (resp. $r_{z,t}$) la relation (ou rapport) liant x à y (resp. z à t), l'analogie peut être simplement définie à l'aide d'une similarité (notée $s(.,.)$) entre ces rapports, i.e. :

$$a(x : y :: z : t) = s(r_{x,y}, r_{z,t}),$$

où $a(.)$ représente une mesure de la validité de la proportion ; cette mesure sera désignée par le terme *analogicité*. Cette approche est en accord avec la conception

euclidienne de l'analogie en tant que « similarité de raisons ». Elle se retrouve également chez Lepage, qui donne à la similarité de relations le nom de *conformité*. L'analogie est alors [1]

> une conformité de rapport entre objets du même type.
>
> Lepage (2003)

L'objectif de Turney & Littman (2005) est de qualifier la relation sémantique entre un nom et un modifieur (telle que *valise* et *roulettes* dans *valise à roulettes*), de façon à rendre plus simples d'autres tâches : dans une application de traduction automatique, il peut être utile de remplacer une paire nom-modifieur par une paraphrase plus facile à traduire ; dans un contexte d'extraction d'information ou de désambiguïsation sémantique la connaissance d'une telle relation permet de détecter plus facilement le rôle de chaque terme.

Turney & Littman se placent dans un cadre d'apprentissage supervisé : à partir de paires nom-modifieur étiquetées par des classes de relations sémantiques, il s'agit de classer des nouvelles paires. Pour résoudre ce problème de classification, le modèle proposé utilise l'algorithme des k-ppv ; celui-ci nécessite de savoir comparer des paires de termes, conduisant à considérer des proportions analogiques entre quatre termes.

cf. Sec. 2.2.1, p. 24

Selon cette conception, deux ingrédients sont nécessaires pour qualifier une analogie : (i) l'expression des relations entre objets, (ii) la donnée d'une similarité entre ces relations.

De façon à exprimer la relation sémantique entre mots, Turney & Littman font intervenir un modèle vectoriel. Dans celui-ci, la relation entre deux mots prend la forme d'un vecteur dont les composantes sont caractérisées par des patrons syntagmatiques du type « *x et y* », « *x avec y* », « *x pour y* », etc. Pour un couple de mots (x, y) donné, la valeur de chaque composante est déterminée par un comptage fréquentiel dans un corpus : pour renseigner la composante « *x à y* » du couple *(valise, roulettes)*, on compte le nombre d'occurrences de l'expression *valise à roulettes* dans un corpus. Ce vecteur représente d'une certaine façon la *signature* de la relation liant x à y. Pour qualifier la proportion $x : y :: z : t$, il suffit de calculer de la sorte les vecteurs $r_{x,y}$ et $r_{z,t}$, et d'appliquer une simple similarité en cosinus, i.e. :

$$a(x : y :: z : t) = cos(\theta_{r_{x,y},r_{z,t}}) = \frac{\langle r_{x,y},\ r_{z,t}\rangle}{\|r_{x,y}\|\|r_{z,t}\|}.$$

Cette vision est à rapprocher du modèle de Plate (2000), déjà évoqué. Rappelons que ce dernier code les relations entre objets sous une forme vectorielle, à l'aide de HRR (*Holographic Reduced Representation*). Dans ce contexte, une similarité entre relations se résume également à une similarité en cosinus.

cf. Sec. 2.1.3, p. 20

Le modèle de Turney & Littman prend donc la forme d'un apprentissage paresseux traditionnel, dans lequel les objets considérés sont des vecteurs codant une relation sémantique entre deux termes. Ainsi, l'apport principal de leur approche

1. Notons que, pour Lepage, cette conformité est pratiquement synonyme d'égalité, ce qui revient à considérer la distance discrète ($d(x, y) = 0$ si $x = y$ et 1 sinon), définissable sur tout ensemble.

réside non pas dans le mécanisme d'apprentissage, mais dans le codage vectoriel des relations sémantiques.

Le modèle d'Hathout Hathout (2001) traite le problème de l'acquisition automatique de liens morphologiques, présenté comme une étape préalable à la constitution de ressources de langue générale pour le français (Hathout *et al.*, 2002). Le modèle proposé permet de structurer le lexique, en identifiant : (i) des paires de termes liés morphologiquement, (ii) des relations sémantiques entre ces paires.

Une analogie entre quatre termes (x, y, z, t) s'opère aux deux niveaux graphique [2] et sémantique. Tout d'abord, les paires (x, y) et (z, t) impliquées dans une analogie sont telles que $x_g = r_1 s_1$, $y_g = r_1 s_2$, $z_g = r_2 s_3$, et $t_g = r_2 s_4$ (α_g représente la graphie de α) : cette contrainte assure que x_g et y_g (ainsi que z_g et t_g) partagent un préfixe commun (dans la pratique on impose en effet une taille minimale pour r_1 et r_2). Le couple (s_1, s_2) (resp. (s_3, s_4)) représente la *signature* de la relation graphique entre x_g et y_g (resp. z_g et t_g). Ensuite, il s'agit de vérifier (à l'aide d'un dictionnaire de synonymes) que x est proche sémantiquement de z, et de même pour y et t. Il est possible de construire de la sorte un graphe dont les nœuds sont des couples de termes morphologiquement liés et les arcs l'indice d'une proximité sémantique, conduisant à une représentation structurée du lexique.

Hathout distingue plusieurs types de proportions analogiques.

- Les proportions *strictes* sont telles que $(s_1, s_2) = (s_3, s_4)$; ce sont les proportions les plus « régulières ». Exemple : *vénérer* : *vénération* :: *adorer* : *adoration*. Une signature impliquée dans une proportion stricte est dite stricte également.
- Dans une proportion *lâche-1*, les signatures (s_1, s_2) et (s_3, s_4) sont toutes les deux strictes, mais néanmoins différentes. Un exemple d'une telle proportion est *changer* : *changement* :: *permuter* : *permutation*, car il est possible de trouver d'une part la proportion *changer* : *changement* :: *remplacer* : *remplacement* (signature stricte *er/ement*), et d'autre part la proportion *commuter* : *commutation* :: *permuter* : *permutation* (signature stricte *er/ation*).
- Les proportions *lâche-2* et *lâche-3* sont également définies, qui comportent respectivement une ou aucune signature stricte (par ex. *réunir* : *réunion* :: *mélanger* : *mélange* et *bon* : *bonté* :: *fort* : *force*).

De cette étude nous retiendrons en particulier que les proportions *lâche-2* et *lâche-3* ne constituent pas une contribution forte à la structuration (morphologique) du lexique : les « piliers » principaux de celui-ci sont formés par les proportions *strictes*, piliers que les proportions *lâche-1* permettent de relier.

> Les spécificités sémantiques et phonologiques des affixes expliquent l'importance particulière des analogies strictes. [...] Ces dernières s'imposent ainsi comme les analogies canoniques [...] et comme le principal axe de structuration paradigmatique du lexique par la morphologie constructionnelle. [...] Les schémas stricts constituent des points (en fait des sous-graphes) de référence par rapport auxquels se définit le sens de la plupart des schémas.
>
> Hathout (2001)

2. Des informations morpho-syntaxiques additionnelles sont également prises en considération.

Signalons enfin que cette structuration de données linguistiques correspond à un cadre d'apprentissage automatique qualifié de *non-supervisé* : aucune « sortie » (information de supervision) n'est associée aux objets rencontrés ; l'objectif est alors de repérer ou de découvrir des « formes » et des « structures » dans les données. Le modèle de Hathout ne prédéfinit pas de classes sémantiques ; celles-ci « émergent » de la procédure d'apprentissage et prennent la forme de zones denses dans le graphe construit.

Le modèle de Claveau & L'Homme Claveau & L'Homme (2005) étudient le problème de la structuration de terminologie. Ils cherchent à découvrir des unités terminologiques morphologiquement liées et à prédire le lien sémantique qui les unit. Les lexiques de spécialité sont principalement visés.

L'apprentissage considéré dans cette approche est supervisé : il s'agit d'exploiter des paires de termes ; à chaque paire est associée une relation sémantique explicite [3]. Chaque paire est caractérisée par une signature graphique : analyser une nouvelle paire consiste à comparer sa signature graphique à celles des paires connues et à en déduire sa relation sémantique à l'aide de l'algorithme des k-ppv. Ce modèle repose, comme celui de Turney & Littman, sur un apprentissage supervisé paresseux impliquant en entrée des signatures et en sortie un codage explicite des relations sémantiques liant deux termes. Par ailleurs, il partage avec le modèle d'Hathout la contrainte que les paires constituant la base d'apprentissage doivent présenter des similarités graphiques (longue sous-chaîne de symboles commune) et sémantiques.

cf. Sec. 2.2.1, p. 24

L'hypothèse sur laquelle cette étude repose peut être formulée de la façon suivante : des signatures graphiques identiques conduisent à des relations sémantiques identiques.

> En nous restreignant à des domaines de spécialité particuliers, nous considérons que des liens morphologiques réguliers trahissent également des liens sémantiques réguliers.
>
> Claveau & L'Homme (2005)

En utilisant la terminologie proposée par Hathout, nous dirons que les proportions exploitées sont préférentiellement les proportions strictes (signatures graphiques identiques) et que celles-ci peuvent constituer (en particulier pour les domaines de spécialité) de bons indicateurs de liens sémantiques.

Du codage de relations aux transformations Les approches étudiées précédemment partagent une conception de la proportion analogique en tant que similarité de relations. Cette vision offre une réponse à la question suivante :

– À quel point le quadruplet (x, y, z, t) forme-t-il une proportion analogique ?

En fixant un seuil, il est également possible de répondre à : le quadruplet (x, y, z, t) forme-t-il une proportion analogique ?

3. Ces relations sémantiques sont déterminées à partir des fonctions lexicales de Mel'cuk.

En revanche, la donnée seule d'une similarité de relations ne permet pas de répondre au problème de la résolution d'équation analogique, au cœur de nos mécanismes d'apprentissage par analogie.

Pour pouvoir résoudre des équations analogiques, une évolution du modèle consiste à coder non seulement des relations mais des « transformations ». En notant $\sigma_{x,y}$ la transformation permettant de « passer » de x à y ($\sigma_{x,y}(x) = y$), si l'ensemble des solutions $S(x : y :: z :?)$ de l'équation $x : y :: z :?$ est donné par :

$$S(x : y :: z :?) = argmax_t s(\sigma_{x,y}, \sigma_{z,t}),$$

alors il est possible d'exploiter des hypothèses telles que :

$$argmax_t s(\sigma_{x,y}, \sigma_{z,t}) \simeq \sigma_{x,y}(z).$$

La signature graphique évoquée plus haut (et qui se retrouve par ailleurs dans de nombreux travaux en acquisition de la morphologie), présentée initialement comme modélisant une relation entre deux formes graphiques, peut également être vue comme un codage de transformation. En effet, la signature $(er/ation)$ « appliquée » à *vénérer* donne *vénération* ; elle correspond à la procédure : « supprimer le suffixe *er* et ajouter le suffixe *ation* ». Il n'est pas possible d'établir cette correspondance entre relations et transformations dans le cas général, comme le montre par exemple le modèle de Turney & Littman.

Cette approche « transformationnelle » est illustrée par le système ANALOGY, proposé par Evans (1968). Dans la modélisation des transformations permettant d'agir sur des figures géométriques, Evans introduit trois opérations élémentaires : (i) l'insertion d'objet, (ii) la suppression d'objet, (iii) la substitution d'objet.

cf. Sec. 2.1.2, p. 17

Dans le contexte des chaînes de symboles, ces opérations sont introduites pour calculer des *distances d'édition* (Levenshtein, 1965 ; Wagner & Fischer, 1974) : la distance d'édition entre deux chaînes de symboles a et b est définie par le nombre minimal d'opérations permettant de « transformer » a en b. Par exemple, la distance entre les chaînes *criera* et *trieur* est de 3 : substitution de c par t, insertion de u et suppression de a.

Le modèle de Miclet, Delhay et Lepage Le modèle de Delhay & Miclet (2005) traite le cas des chaînes de symboles et peut être positionné en quelque sorte dans la continuité de celui d'Evans. En effet, ils définissent la relation ou la transformation entre deux chaînes x et y par la *trace* $\sigma_{x,y}$ résultant du calcul de la distance d'édition entre x et y. La trace code la succession des opérations élémentaires à appliquer à une chaîne pour obtenir un autre chaîne. Par exemple, la trace associée au couple (*criera*, *trieur*) peut être notée $s_{c,t} - - - i_u - d_a$, où $s_{x,y}$, i_x, d_x, et $-$ représentent respectivement la substitution du symbole x par y, l'insertion du symbole x, la délétion du symbole x et la simple recopie du symbole courant (effectuée à coût nul).

La trace correspond à une séquence d'opérations et peut, elle aussi, être codée à l'aide d'une chaîne de symboles. Il est donc tout à fait possible de définir une dis-

tance d'édition entre traces, ce qui permet de définir une *dissemblance analogique* :

$$DA(x, y, z, t) = \Delta(\sigma_{x,y}, \sigma_{z,t}),$$

où Δ correspond à la distance d'édition entre traces [4].

La transformation représentée par la trace n'est pas applicable à une chaîne quelconque ; par exemple, il n'est pas possible d'appliquer la trace $s_{c,t} - - - i_u - d_a$ à la chaîne *peur* car cette dernière comporte 4 lettres et ne débute pas par la lettre *c*. Il n'est donc pas possible d'exploiter directement une identité telle que $t = \sigma_{x,y}(x)$. Pour résoudre ce problème, Delhay & Miclet se reposent simultanément sur la trace $\sigma_{x,y}$ entre x et y et la trace $\sigma_{x,z}$ entre x et z, de manière à produire t. Notons finalement que l'algorithme proposé par Lepage (1998) implique également [5] une exploitation simultanée des deux traces $\sigma_{x,y}$ et $\sigma_{x,z}$.

4. Plus précisément, ce qui est calculé est la distance minimale entre toutes les traces possibles (y compris non-optimales).

5. Dans ce contexte, l'utilisation du terme *également* a pour objet des considérations uniquement pédagogiques puisque le modèle de Lepage est antérieur à celui de Miclet et Delhay.

ANNEXE

B

Alanis

Dans le cadre de ce travail, nous avons développé un logiciel, baptisé ALANIS (*A Learning by ANalogy Inferencer for Structured Data* : un apprenti à base d'analogies pour l'apprentissage de données structurées). Celui-ci implante (i) la méthode d'apprentissage automatique APPA (cf. section 3.2), (ii) les algorithmes liés aux proportions analogiques pour un certain nombre de structures algébriques (cf. chapitre 4), (iii) des mécanismes de filtrage et de pondérations de solutions (cf. section 5.1.2). Nous décrivons dans cette annexe l'architecture globale du système, ses propriétés telles que la généricité, les formats d'entrées/sorties ainsi que des questions de complexité. Notons que le langage principal utilisé pour développer ALANIS est le langage C++ ; un certain nombre de pré-traitements et post-traitements sont effectués à l'aide du langage PYTHON.

Résolution d'équations analogiques : ALANIS et la généricité Les solutions proposées au problème de la définition de la notion de proportion analogique s'insérent dans un cadre algébrique unifié. Pour pouvoir rendre compte au mieux de cette propriété, nous avons cherché à rendre l'implantation de la résolution d'équations analogiques la plus générique possible.

cf. Sec. 4.2, p. 91

De façon à atteindre cet objectif de généricité, l'implantation choisie repose sur le *design-pattern* (Gamma *et al.* (1999)) sous-jacent à la bibliothèque VAUCANSON [1] (Lombardy *et al.*, 2004). Ce *design-pattern* a initialement été conçu de manière à manipuler des automates généralisés (voir par ex. Sakarovitch (2003)), c'est-à-dire des automates dont les transitions peuvent être étiquetées non seulement par des lettres (symboles d'un alphabet Σ), mais également pas des mots (éléments du monoïde libre $\Sigma^\star$) ou encore des séries (applications d'un monoïde libre $\Sigma^\star$ vers un semi-anneau $(A, +, \times)$). Dans ce contexte, il est important de différencier la structure algébrique (i.e. l'entité mathématique) associée aux éléments manipulés (monoïde, semi-anneau, etc.) de la structure informatique concrète utilisée pour représenter ces éléments (entiers, flottants, tableaux, chaînes de caractères, listes, etc.).

1. Disponible depuis `http://www.lrde.epita.fr/cgi-bin/twiki/view/Vaucanson/Vaucanson`.

Dans un contexte de programmation orientée objet, la technique habituellement utilisée pour traiter ce type de situation est le polymorphisme : un algorithme est défini pour une forme générale de base B et peut s'appliquer aux différentes formes (héritant de B) que peut prendre un objet. Si nécessaire, il est possible de redéfinir cet algorithme pour des formes particulières plus spécifiques que B.

Cependant, le simple polymorphisme ne permet pas de rendre compte des deux hiérarchies indépendantes que représentent les structures algébriques (une série est un cas particulier de monoïde, lui même un cas particulier de semigroupe, etc.) et les types concrets (B est un cas particulier de A, etc.). Il est alors difficile de spécialiser les algorithmes selon ces deux axes de manière indépendante. En outre, le polymorphisme présente un autre défaut majeur : la sélection des algorithmes à l'execution représente un coût (d'indirection) non négligeable lorsque les algorithmes sont fortement sollicités.

Signalons enfin que le rôle premier du polymorphisme consiste à pouvoir gérer des objets réellement polymorphes, c'est-à-dire dont la forme est changeante au cours de l'execution du programme. Dans le cas d'ALANIS, le contexte algébrique (la nature des objets) est fixé pour toute la durée d'execution du programme et on souhaite pouvoir éviter le coût lié à la séléction des algorithmes.

Le *design-pattern* de VAUCANSON repose précisément sur un « polymorphisme statique » qui permet le partage et la réutilisation du code, la séparation des structures algébriques de leurs représentations concretes, la spécialisation des algorithmes selon ces deux axes de manière indépendante et ceci sans perte d'efficacité à l'execution (Régis-Gianas & Poss, 2003).

Ce design-pattern forme le cœur du moteur d'apprentissage : le contexte algébrique (qui donne un sens à la notion de proportion analogique) est totalement indépendant des types concrets utilisés pour représenter les éléments manipulés. Par exemple, les types `double` et `float` (flottants) peuvent être utilisés pour modéliser des éléments de $(\mathbb{R}, +)$ ou $(\mathbb{R}^*, \times)$ et des `int` (entiers) peuvent être utilisés pour modéliser des éléments de $(\mathbb{Z}, +)$. Dans tous les cas, si l'on considère que la structure algébrique sous-jacente est un groupe abélien, le même algorithme peut être exploité pour gérer toutes ces situations. En outre, il est possible de spécialiser un algorithme relativement à l'implantation : il est alors possible d'effectuer des optimisations additionnelles qui tiennent compte de la nature spécifique des types considérés.

Architecture d'ALANIS Le moteur d'apprentissage au cœur d'ALANIS repose en premier lieu sur la définition de deux contextes correspondant respectivement à l'espace d'entrée (type `input_analogical_framework_t`) et à l'espace de sortie (`output_analogical_framework_t`). Ces contextes sont caractérisées par la donnée d'une structure algébrique (par ex. `inputs_t`) et d'un type concret associé à son implantation (par ex. `input_value_t`). Les deux structures algébriques (en entrée et en sortie) donnent un sens à la notion de proportion analogique, telle que définie dans le chapitre 4. Par exemple, pour la tâche de prononciation, puisque les espaces

cf. Sec. 5.2, p. 127

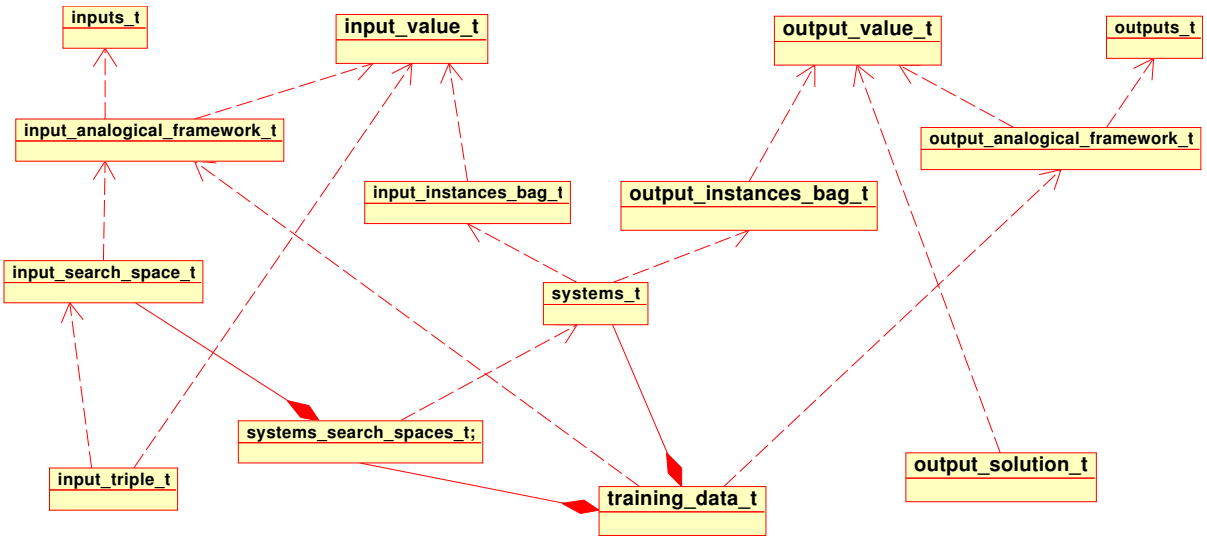


FIG. B.1 – Architecture d'ALANIS

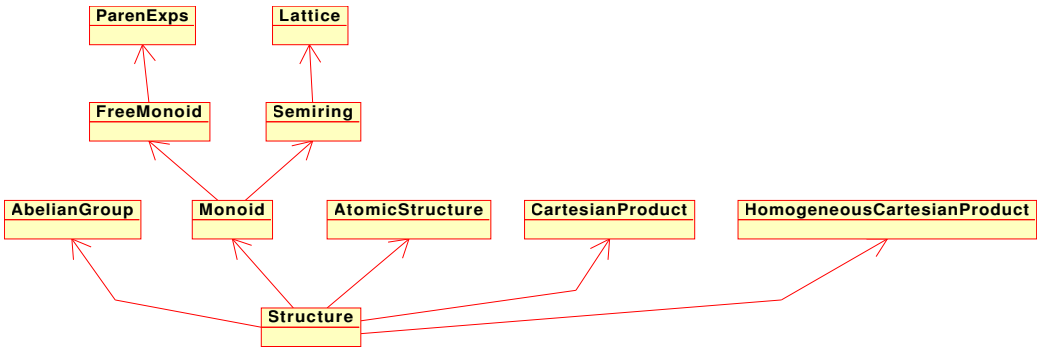


FIG. B.2 – Structures algébriques utilisées

d'entrée et de sortie sont des monoïdes libres, on a `inputs_t = FreeMonoid` et `outputs_t = FreeMonoid`. Pour la tâche d'analyse flexionnelle, l'espace de sortie est l'ensemble des structures de traits plates, modélisable par un produit cartésien homogène (E^n) : dans ce cas, `outputs_t = HomogenousCartesianProduct`. Pour la tâche d'analyse dérivationnelle, l'espace de sortie est l'ensemble des expressions parenthésées codant des arbres : `outputs_t = ParenExps`. Les types concrets (`input_value_t` et `output_value_t`) correspondent aux types C++ utilisés pour implanter ces structures, par exemple des `std::string` ou des `std::vector`.

cf. Sec. 5.3, p. 137

cf. Sec. 5.4, p. 140

Tous les exemples sont stockés dans des « sacs » qui assignent un identifiant unique à toute valeur recontrée : `input_instances_bag` pour les valeurs dans l'espace d'entrée et `output_instances_bag` pour les valeurs dans l'espace de sortie. L'association entre une valeur d'entrée et une valeur de sortie s'effectue à l'aide d'un `systems_t` : celui-ci représente donc une relation au sens mathématique du terme. Ce `systems_t` est en réalité un ensemble de `system_t`, chaque `system_t`

correspondant à une « catégorie » telle que présentée dans la section 5.1.3.

L'ensemble d'apprentissage (type `training_data_t`) est constitué d'un tel ensemble de catégories (`systems_t`) ainsi que d'un espace de recherche représenté par un objet de type `systems_search_spaces_t` lui permettant d'effectuer la recherche des triplets (`input_triple_t`) dans l'espace d'entrée. Enfin, chaque solution à une équation analogique est du type `output_solution_t`.

Formats d'entrées/sorties Le format d'entrée pour les fichiers utilisés par ALANIS est tabulaire (cf. figure B.3). Chaque exemple de la base d'apprentissage est présenté au système sous la forme d'une ligne composée de trois champs : `input`, `system_id` et `output`, où `input` (resp. `output`) représente la valeur de l'exemple dans l'espace d'entrée (resp. de sortie) et `system_id` un identifiant du « système » ou de la « catégorie » relatifs à l'exemple. Toute entrée ambiguë donne lieu à plusieurs lignes de la sorte. Les chaînes `input` et `output` sont tout d'abord converties dans les types adéquats (`input_value_t` et `output_value_t`) à l'aide d'adapteurs, pour être ensuite intégrées dans les structures nécessaires au déroulement de l'apprentissage : `input_instances_bag`, `output_instances_bag`, `systems_t`, `training_data_t` et `systems_search_spaces_t`. Pour les exemples de la base de test, le fichier est constitué d'une ligne par entrée présentée.

cf. Sec. 5.1.3, p. 126

```
absence of mind:134:{bsHsQvm2nd
absence of mind:134:{bsHs@vm2nd
absence of mind:134:{bs@nsQvm2nd
absence of mind:134:{bs@ns@vm2nd
absences:133:{bsHsIz
absences:133:{bs@nsIz
absent:135:{bsEnt
absent:135:@bsEnt
absent:136:{bsHt
absent:136:{bs@nt
```

FIG. B.3 – Exemple de fichier d'entrée

ALANIS propose plusieurs solutions pondérées pour une entrée donnée. De manière à faciliter les post-traitements et notamment le filtrage de ces solutions, celles-ci sont présentées sous un format XML, aisément ré-exploitable (cf. figure B.4). Toute l'information nécessaire à ce filtrage est présente dans le fichier : la valeur des solutions ainsi que leurs pondérations. En outre, les informations initiales sur l'entrée sont reproduites dans la sortie pour simplifier l'étape de comparaison : il s'agit des sorties attendues, mais également de descripteurs additionnels permettant de ventiler les solutions selon certains critères (partie du discours, caractère ambigu, etc.). Un programme spécifique permet de parser de tels fichiers XML [2] pour en extraire ces informations, classer les solutions et les filtrer à l'aide des différentes stratégies de filtrage proposées dans la section 5.1.2.

2. Nous avons pour cela utilisé la bibliothéque LIBXML : `http://www.xmlsoft.org/`.

```
<instance>
<actual><wordform val="accumulation" id="454" >
<pronunciation val="@kjumj@l1SH" descriptors="N,2,0,S,100" />
<pronunciation val="@kjumjUl1SH" descriptors="N,2,0,P,100" />
</wordform></actual>
<induced>
<input val="accumulation" />
<output n_triples="166" >
<solution val="@kjumj@l1S7n" weight="2.5" />
<solution val="@kjumj@l1S@n" weight="1.625" />
<solution val="@kjumj@l1SH" weight="43.0625" />
<solution val="@kjumj@l1SHR" weight="2.875" />
<solution val="@kjumj@l1SHr" weight="0.03125" />
<solution val="@kjumj@l1SIn" weight="0.25" />
<solution val="@kjumj@l1Sj@n" weight="2.5" />
<solution val="@kjumj@l1s7n" weight="1" />
<solution val="@kjumj@l1sj@n" weight="1" />
<solution val="@kjumjUl1S7n" weight="2.5" />
<solution val="@kjumjUl1S@n" weight="1.625" />
<solution val="@kjumjUl1SH" weight="43.0625" />
<solution val="@kjumjUl1SHR" weight="2.875" />
<solution val="@kjumjUl1SHr" weight="0.03125" />
<solution val="@kjumjUl1SIn" weight="0.25" />
<solution val="@kjumjUl1Sj@n" weight="2.5" />
<solution val="@kjumjUl1s7n" weight="1" />
<solution val="@kjumjUl1sj@n" weight="1" />
</output>
</induced>
</instance>
```

FIG. B.4 – Exemple de fichier de sortie

Questions de complexité Nous avons proposé des algorithmes concernant la résolution d'équations analogiques dans la section 4.1.3. Les solveurs à base d'automates présentent l'ensemble des solutions à une équation analogique sous la forme d'un automate non-déterministe. Lorsqu'il s'agit de vérifier qu'un mot appartient à l'ensemble des solutions, une telle représentation est suffisante. En effet, l'opération de vérification d'appartenance d'un mot x à un langage représenté par un automate A s'effectue en $O(|x| \times |A|)$.

cf. Sec. 4.1.3, p. 83

En revanche, lorsqu'il s'agit d'énumérer l'ensemble des solutions, il faut explorer intégralement l'automate, opération s'effectuant à une complexité potentiellement exponentielle (relativement à $|A|$).

Dans la suite, nous démontrons la proposition suivante.

Proposition 16. Si le degré maximal d'une proportion est limité à d, alors l'ensemble des solutions d'une équation analogique $x : y :: z :?$ peut être calculé en $O((|x| \times |y| \times |z|)^{d-1})$.

cf. Sec. 4.1.2, p. 79

Démonstration. Selon la définition 4, identifier une proportion analogique entre 4 mots x, y, z et t revient à factoriser x (resp. y, z, t) en facteurs $x_1 \dots x_n = x$ (resp. $y_1 \dots y_n = y$, $z_1 \dots z_n = z$ et $t_1 \dots t_n = t$) et à aligner les facteurs obtenus (i.e. $\forall i \in [\![1, n]\!]$, $(x_i, t_i) \in \{(y_i, z_i), (z_i, y_i)\}$).

Segmenter un mot en d facteurs consiste à définir $d - 1$ points de coupure. De manière à résoudre l'équation $x : y :: z :?$, on identifie tout d'abord $d - 1$ points de coupure dans x, y et z. Il existe $(|x + 1| \times |y + 1| \times |z + 1|)^{d-1})$ façons de définir de telles coupures. Une fois ces points de coupure identifiés, il existe des solutions si et seulement si $\forall i \in [\![1, d]\!]$, $x_i = y_i$ ou $x_i = z_i$. Dans ce cas, les solutions s'obtiennent directement et sont de la forme $t = t_1 \dots t_d$ avec $t_i = z_i$ si $x_i = y_i$ et $t_i = y_i$ si $x_i = z_i$. Il y a au plus 2^d combinaisons de la sorte, ce qui justifie la complexité globale suivante :

$$(|x + 1| \times |y + 1| \times |z + 1|)^{d-1} \times 2^d = O((|x| \times |y| \times |z|)^{d-1}).$$

□

Bibliographie

AAMODT A. & PLAZA E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, 7(1):39–59. 14

AHA D.W. (1997). Editorial. *Artificial Intelligence Review*, 11(1-5):7–10, special Issue on Lazy Learning. 28, 54

AHA D.W., KIBLER D. & ALBERT M.K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1):37–66. 28, 54, 58

ARISTOTE. *Poétique*. Les Belles Lettres, Paris, (traduction de J. Hardy, 1990). 10

ARMENGOL E. & PLAZA E. (2001). Lazy induction of descriptions for relational case-based learning. In L.D. Raedt & P. Flach (eds.), *Proceedings of the 12thEuropean Conference on Machine Learning*, vol. 2167 de *Lecture Notes in Computer Science*, p. 13–24, Springer-Verlag. 30

ARTHERN P.J. (1978). Machine translation and computerized terminology systems: a translator's viewpoint. In *Translating and the computer: proceedings of a seminar, London, 14thnovember 1978*, p. 77–108, North-Holland, Amsterdam, The Netherlands. 37

BENGIO Y. & FRASCONI P. (1996). Input/Output HMMs for sequence processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249. 36

BLOOMFIELD L. (1970). *Le Langage*. Payot, 1er édition. 32

BOUSQUET O. (2003). Statistical learning theory. In *Machine Learning Summer School*, Tuebingen, Germany. 63, 64

BURNAGE G. (1990). CELEX: a guide for users. Rapport technique, University of Nijmegen, Center for Lexical Information. 128

BURRUS N., DURET-LUTZ A., GERAUD T., LESAGE D. & POSS R. (2003). A static C++ object-oriented programming (SCOOP) paradigm mixing benefits of traditional OOP and generic programming. In *Proceedings of the Workshop on Multiple Paradigm with OO Languages (MPOOL '03)*, Anaheim, CA. 120

CARL M. & WAY A. (eds.) (2003). *Recent Advances in Example-Based Machine Translation*, vol. 21 de *Text, Speech and Language Technology*. Kluwer Academic Publishers. 38

CARPENTER B. (1992). *The Logic of Typed Feature Structures*. N° 32 In Cambridge Tracts in Theoretical Computer Science, Cambridge University Press. 98

CARRERAS X., MÀRQUEZ B.L. & CASTRO C.J. (2005). Filtering-ranking perceptron learning for partial parsing. *Machine Learning*, 60(1-3):41–71. 48

CHALMERS D.J., FRENCH R.M. & HOFSTADTER D.R. (1995). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. In D.R. Hofstadter & the Fluid Analogies Research Group (eds.), *Fluid Concepts and Creative Analogies*, chap. 4, p. 169–193, Basic Books, New York, NY. 22

CHARNIAK E. (1996). *Statistical Language Learning*. The MIT Press, Cambridge, MA. 35

CHARNIAK E. & JOHNSON M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, p. 173–180, Association for Computational Linguistics, Ann Arbor, Michigan. 48

CHOMSKY N. (1957). *Syntactic Structures*. Mouton, The Hague. 33

CLAVEAU V. & L'HOMME M.C. (2005). Apprentissage par analogie pour la structuration de terminologie - utilisation comparée de ressources endogènes et exogènes. In *Conférence TIA-2005*. 151, 154

COLLINS M. & DUFFY N. (2001). Convolution kernels for natural language. In T. Dietterich, S. Becker & Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems*, vol. 14, The MIT Press, Cambridge, MA. 48, 51

COLLINS M. & KOO T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69. 48, 51

COMON H., DAUCHET M., GILLERON R., JACQUEMARD F., LUGIEZ D., TISON S. & TOMMASI M. (1997). Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, release October, 1st2002. 110

CORNUÉJOLS A. & MICLET L. (2002). *Apprentissage Artificiel: Concepts et Algorithmes*. Eyrolles. 24, 63, 121

COVER T. & HART P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27. 24

DAELEMANS W., GILLIS S. & DURIEUX G. (1997*a*). Skousen's analogical modeling algorithm: A comparison with lazy learning. In D. Jones & H. Somers (eds.), *New Methods in Language Processing*, p. 3–15, University College Press, London. 37

DAELEMANS W. & VAN DEN BOSCH A. (2001). Treetalk: Memory-based word phonemisation. In R.I. Damper (ed.), *Data-Driven Techniques in Speech Synthesis*, vol. 9 de *Telecommunications Technologies & Applications Series*, chap. 7, p. 149–172, Kluwer Academic Publishers, Boston. 36, 46

DAELEMANS W. & VAN DEN BOSCH A. (2005). *Memory-Based Language Processing*. Studies in Natural Language Processing, Cambridge University Press. 36, 46

DAELEMANS W., VAN DEN BOSCH A. & WEIJTERS T. (1997*b*). IGTree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review, special issue on Lazy Learning*, 11:407–423. 58

DAELEMANS W., ZAVREL J., BERCK P. & GILLIS S. (1996). MBT: A memory-based part of speech tagger-generator. In E. Ejerhed & I. Dagan (eds.), *Proceedings of the 4th Workshop on Very Large Corpora*, p. 14–27. 36, 46

DAELEMANS W., ZAVREL J., VAN DER SLOOT K. & VAN DEN BOSCH A. (2004). Timbl: Tilburg memory based learner, version 5.1, reference guide. Rapport technique 04-02, ILK. 47, 120, 130

DASARATHY B.V. (ed.) (1990). *Nearest neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA. 58, 127

DASARATHY B.V. (1991). *Nearest Neighbor Pattern Classification Techniques*. IEEE Computer Society Press. 24

DASARATHY B.V., SÁNCHEZ J.S. & TOWNSEND S. (2000). Nearest neighbour editing and condensing tools - synergy exploitation. *Pattern Analysis and Applications*, 3(1):19–30. 58, 127

DE SAUSSURE F. (1916). *Cours de linguistique générale*. Payot, Lausanne et Paris, 2^{e} édition. 32, 59, 68, 69, 74

DELAHAYE J.P. (1999). *Information, complexité et hasard*. Hermès, Paris, 2^{e} édition. 109

DELHAY A. & MICLET L. (2005). Analogie entre séquences. définition, calcul et utilisation en apprentissage supervisé. *Revue d'Intelligence Artificielle*, 19(4-5):683–712. 155, 156

DEMPSTER A.P., LAIRD N.M. & RUBIN D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 39(1):1–38. 130

DENIS F. (2000). Apprentissage automatique : des modèles formels aux applications. Habilitation à diriger des recherches, Université des Sciences et Technologies de Lille. 68

DEVIJVER P. & KITTLER J. (1980). On the edited nearest neighbor rule. In *Proceedings of the 5th International Conference on Pattern Recognition*, p. 72–80, Miami Beach, Florida. 58

DIETTERICH T.G. (1997). Approximate statistical tests for comparing supervised classification learning algorithms. Rapport technique, Department of Computer Science, Oregon State University. 121

DIETTERICH T.G. (2002). Machine learning for sequential data: A review. In T. Caelli, A. Amin, R.P.W. Duin, M.S. Kamel & D. de Ridder (eds.), *Structural, Syntactic, and Statistical Pattern Recognition*, vol. 2396 de *Lecture Notes in Computer Science*, p. 15–30, Springer-Verlag. 36

DIETTERICH T.G., LATHROP R.H. & LOZANO-PÉREZ T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71. 30

DUDA R.O., HART P.E. & STORK D.G. (2000). *Pattern Classification*. Wiley Interscience, 2^{e} édition. 24

DUNBAR K. (2000). What scientific thinking reveals about the nature of cognition. In K. Crowley, C. Schunn & T. Okada (eds.), *Designing for Science: Implications from Everyday, Classroom, and Professional Settings*, p. 115–140, Lawrence Erlbaum Associates, Mahwah, NJ. 10

DUPONT P. & MICLET L. (1998). Inférence grammaticale régulière : fondements théoriques et principaux algorithmes. Rapport technique 3449, INRIA. 63

ELIASMITH C. & THAGARD P. (2001). Integrating structure and meaning: a distributed model of analogical mapping. *Cognitive Science*, 25(2):245–286. 21

EMDE W. & WETTSCHERECK D. (1996). Relational instance based learning. In L. Saitta (ed.), *Proceedings of the 13thInternational Conference on Machine Learning*, p. 122–130, Morgan Kaufmann Publishers. 30

EVANS T. (1968). A program for the solution of a class of geometric analogy intelligence test questions. In M. Minsky (ed.), *Semantic Information Processing*, p. 271–353, MIT Press, Cambridge, MA. 17, 155

FALKENHAINER B., FORBUS K.D. & GENTNER D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41:1–63. 8, 19, 20

FIX E. & HODGES J. (1951). Discriminatory analysis, non-parametric discrimination: consistency properties. Rapport technique 21-49-004, 4, US Air Force, School of Aviation and Medicine. 24

FORBUS K.D., FERGUSON R.W. & GENTNER D. (1994). Incremental structure-mapping. In A. Ram & K. Eiselt (eds.), *Proceedings of the 16thAnnual Conference of the Cognitive Science Society*, p. 313–318, Lawrence Erlbaum Associates. 20

FORBUS K.D., GENTNER D. & LAW K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19(2):141–205. 20

FRADIN B. (2003). *Nouvelles approches en morphologie*. Presses Universitaires de France, Paris, France. 65, 68, 75

FRENCH R.M. (2002). The computational modeling of analogy-making. *Trends in Cognitive Science*, 6(5):200–205. 16

GAMMA E., HELM R., JOHNSON R. & VLISSIDES J. (1999). *Design patterns : Catalogue de modèles de conception réutilisables*. Vuibert, Paris, France. 157

GÄRTNER T., LLOYD J.W. & FLACH P.A. (2004). Kernels and distances for structured data. *Machine Learning, special issue on Inductive Logic Programming*, 57(3):205–232. 51

GATES G. (1972). The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18(3):431–433. 58

GAZDAR G. & MELLISH C. (1989). *Natural Language Processing in Prolog: An Introduction to Computational Linguistics*. Addison-Wesley, Workingham, UK. 35

GENTNER D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170. 19

GENTNER D., BREM S., FERGUSON R., MARKMAN A., LEVIDOW B., WOLFF P. & FORBUS K. (1997). Analogical reasoning and conceptual change: A case study of Johannes Kepler. *The Journal of the Learning Sciences*, 6(1):3–40. 10

GENTNER D. & FORBUS K.D. (1991). MAC/FAC: A model of similarity-based access and mapping. In *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, p. 504–509, Lawrence Erlbaum Associates, Hillsdale, NJ. 20

GENTNER D., HOLYOAK K.J. & KOKINOV B. (eds.) (2001). *The Analogical Mind: Perspectives from Cognitive Science*. MIT Press, Cambridge, MA. 6, 8, 16

GRUNE D. & JACOBS C.J. (1990). *Parsing Techniques: A Practical Guide*. Ellis Horwood, Chichester, UK. 35

HALL R.P. (1989). Computational approaches to analogical reasoning: a comparative analysis. *Artificial Intelligence*, 39(1):39–120. 8, 16, 29

HART P. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516. 58

HASTIE T., TIBSHIRANI R. & FRIEDMAN J.H. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag. 24

HATHOUT N. (2001). Analogies morpho-synonymiques. Une méthode d'acquisition automatique de liens morphologiques à partir d'un dictionnaire de synonymes. In *Actes de TALN 2001*, p. 223–232, Tours. 151, 153, 154

HATHOUT N., NAMER F. & DAL G. (2002). An experimental constructional database: The mortal project. In P. Boucher (ed.), *Many Morphologies*, Cascadilla, Somerville, MA. 153

HAUSSLER D. (1999). Convolution kernels on discrete structures. Rapport technique UCSC-CRL-99-10, University of California at Santa Cruz. 51

HETTICH S., BLAKE C. & MERZ C. (1998). UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`. 29

HOFSTADTER D.R. & MITCHELL M. (1995). The copycat project: A model of mental fluidity and analogy-making. In D.R. Hofstadter & the Fluid Analogies Research group (eds.), *Fluid Concepts and Creative Analogies*, chap. 5, p. 205–267, Basic Books, New York, NY. 22

HOFSTADTER D.R. & THE FLUID ANALOGIES RESEARCH GROUP (eds.) (1995). *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, New York, NY. 109

HOLYOAK K.J. & THAGARD P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13:295–355. 23

HOLYOAK K.J. & THAGARD P. (1995). *Mental leaps: Analogy in creative thought*. MIT Press, Cambridge, MA. 10

HORVÁTH T., WROBEL S. & BOHNEBECK U. (2001). Relational instance-based learning with lists and terms. *Machine Learning*, 43(1-2):53–80, special issue on inductive logic programming. 30

HUMMEL J.E. & HOLYOAK K.J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104(3):427–466. 23

ITKONEN E. & HAUKIOJA J. (1997). A rehabilitation of analogy in syntax (and elsewhere). In A. Kertész (ed.), *Metalinguistik im Wandel: die kognitive Wende in Wissenschaftstheorie und Linguistik*, p. 131–177, Peter Lang, Frankfurt. 33

KANT E. (1781). *Critique de la raison pure, Théorie transcendantale des éléments*. P.U.F., (traduction de A. Tremesaygues et B. Pacaud, 1967). 22

KASHIMA H. & KOYANAGI T. (2002). Kernels for semi-structured data. In *Proceedings of the 19thInternational Conference on Machine Learning*, p. 291–298, Morgan Kaufmann, San Francisco, CA. 51

KAY M. (1980). The proper place of men and machines in language translation. Rapport technique CSL-80-11, Xerox Palo Alto Research Center, Palo Alto, CA. 37

KAYSER D. (1997). *La représentation des connaissances*. Hermès, Paris, France. 14

KOKINOV B. & FRENCH R.M. (2003). Computational models of analogy-making. In L. Nadel (ed.), *Encyclopedia of Cognitive Science*, p. 113–118, Nature Publishing Group, London. 16

KOLODNER J.L. (1993). *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, CA. 14, 16

LAFFERTY J.D., MCCALLUM A. & PEREIRA F.C.N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18thInternational Conference on Machine Learning*, p. 282–289. 36, 50

LAVIE R.J. (2003). *Le locuteur analogique ou la grammaire mise à sa place*. Thèse de Doctorat, Université de Paris 10, France. 33

LAVRAČ N. & DŽEROSKI S. (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, Chicester, UK. 30

LEAKE D.B. (ed.) (1996). *Case-Based Reasoning: Experiences, Lessons and Future Directions*. MIT Press, Cambridge, MA. 14

LEPAGE Y. (1998). Solving analogies on words: an algorithm. In *Proceedings of COLING-ACL 1998*, vol. 1, p. 728–735, Montréal, Canada. 80, 156

LEPAGE Y. (1999). Open set experiments with direct analysis by analogy. In *Proceedings of the 5thNatural Language Processing Pacific Rim Symposium (NLPRS 1999)*, p. 363–368, Beijing, China. 116, 151

LEPAGE Y. (2000). Languages of analogical strings. In *Proceedings of COLING 2000*, vol. 1, p. 488–494, Saarbrücken, Germany. 59

LEPAGE Y. (2001). Défense et illustration de l'analogie. In *Actes de la 8^{e} Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2001)*, p. 373–377, Tours, France. 33

LEPAGE Y. (2003). De l'analogie rendant compte de la commutation en linguistique. Habilitation à diriger les recherches, Grenoble, France. 33, 75, 77, 152

LEPAGE Y. & DENOUAL E. (2005). The purest EBMT system ever built: no variables, no templates, no training, examples, just examples, only examples. In *Proceedings of the 2ndWorkshop on Example-Based Machine Translation*. 151

LEVENSHTEIN V.I. (1965). Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, english translation in Soviet Physics Doklady, 10(8):707-710, 1966. 155

LI M. & VITÁNYI P. (1997). *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag, New York, 2^{e} édition. 109

LOMBARDY S., RÉGIS-GIANAS Y. & SAKAROVITCH J. (2004). Introducing vaucanson. *Theoretical Computer Science*, 328:77–96. 86, 120, 157

MANNING C.D. & SCHÜTZE H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA. 35

MARCHAND Y. & DAMPER R.I. (2000). A multi-strategy approach to improving pronunciation by analogy. *Computational Linguistics*, 26(2):195–219. 128

MATTHEWS P.H. (1974). *Morphology*. Cambridge University Press, Cambridge. 65

MATTHEWS P.H. (1981). *Syntax*. Cambridge Textbooks in Linguistics, Cambridge University Press. 69, 75

MCCALLUM A., FREITAG D. & PEREIRA F. (2000). Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, p. 591–598, Morgan Kaufmann. 36

MICLET L., BAYOUDH S. & DELHAY A. (2005). Définitions et premières expériences en apprentissage par analogie dans les séquences. In *Conférence d'Apprentissage (CAp '05)*. 80

MICÓ M.L., ONCINA J. & VIDAL E. (1994). A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recognition Letters*, 15(1):9–17. 58

MIKHEEV A. (1997). Automatic rule induction for unknown word guessing. *Computational Linguistics*, 23(3):405–423. 137

MITCHELL M. (2001). Analogy-making as a complex adaptive system. In L.A. Segel & I.R. Cohen (eds.), *Design Principles for the Immune System and Other Distributed Autonomous Systems*, Oxford University Press, New York. 22

MITCHELL T.M. (1980). The need for biases in learning generalizations. Rapport technique CBM-TR-117, Rutgers Computer Science Department Technical Report, reprinted in Readings in Machine Learning, J. Shavlik and T. Dietterich, eds., Morgan Kaufmann, 1990. 63

MITCHELL T.M. (1997). *Machine Learning*. McGraw-Hill. 24, 28, 121

NAGAO M. (1984). A framework of a mechanical translation between japanese and english by analogy principle. In A. Elithorn & R. Banerji (eds.), *Artificial and Human Intelligence*, p. 173–180, North-Holland, Amsterdam, The Netherlands. 38

NEWELL A. (1980). Physical symbol systems. *Cognitive Science*, 4(2):135–183. 17, 21

NEWELL A. & SIMON H.A. (1976). Computer science as empirical inquiry: symbols and search. *Communications of the ACM*, 19(3):113–126. 17, 21

NIPS (2004). Workshop on learning with structured outputs. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*. 51

OCH F.J. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51. 130

PASCAL (2006). Programme thématique : Learning with complex and structured outputs. 1er décembre 2005 - 30 juin 2006. 51

PIRRELLI V. & YVON F. (1999). The hidden dimension: paradigmatic approaches to data-driven natural language processing. *Journal of Experimental and Theoretical Artificial Intelligence, Special Issue on Memory-Based Language Processing*, 11:391–408. 2, 52, 53, 68

PLATE T.A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641. 20

PLATE T.A. (2000). Analogy retrieval and processing with distributed vector representations. *Expert systems*, 17(1):29–40. 20, 152

RABINER L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. 36

RABINER L. & JUANG B. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16. 49

RAMON J. & RAEDT L.D. (1999). Instance based function learning. In S. Džeroski & P. Flach (eds.), *Proceedings of the 9th International Workshop on Inductive Logic Programming*, vol. 1634 de *Lecture Notes in Artificial Intelligence*, p. 268–278, Springer-Verlag, London, UK. 30

RÉGIS-GIANAS Y. & POSS R. (2003). On orthogonal specialization in C++: Dealing with efficiency and algebraic abstraction in Vaucanson. In *Proceedings of the Parallel/High-performance Object-Oriented Scientific Computing (POOSC 2003)*, Darmstadt, Germany. 120, 158

RÉGIS-GIANAS Y. & YVON F. (2004). Dimi. In *Actes des 15^{e} Journées d'Études sur la Parole*, p. 421–425. 34

SAKAROVITCH J. (2003). *Éléments de théorie des automates*. Vuibert, Paris. 84, 105, 157

SATO S. & NAGAO M. (1990). Toward memory-based translation. In *Proceedings of COLING-90*, vol. 3, p. 247–252. 38

SCHMID U., GUST H., KÜHNBERGER K.U. & BURGHARDT J. (2003). An algebraic framework for solving proportional and predictive analogies. In F. Schmalhofer, R. Young & G. Katz (eds.), *Proceedings of the European Conference on Cognitive Science (EuroCogSci 2003)*, p. 295–300, Lawrence Erlbaum, Osnabrück, Germany. 20, 109

SCHÖLKOPF B. & SMOLA A.J. (2002). *Learning with Kernels*. The MIT Press, Cambridge, MA. 51

SHEN L. & JOSHI A.K. (2005). Ranking and reranking with perceptron. *Machine Learning*, 60(1-3):73–96. 48

SKOUSEN R. (1989). *Analogical Modeling of Language*. Kluwer Academic Publishers, Dordrecht. 37

SOMERS H. (1999). Review article: Example-based machine translation. *Machine Translation*, 14:113–157. 38

SOMERS H. (2001). EBMT seen as case-based reasoning. In *Proceedings of MT Summit VIII Workshop on Example-Based Machine Translation*, p. 56–65, Santiago de Compostela, Spain. 38

STROPPA N. & YVON F. (2005). An analogical learner for morphological analysis. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005)*, Ann Arbor, MI. 139, 151

SUMITA E., IIDA H. & KOHYAMA H. (1990). Translating with examples: A new approach to machine translation. In *Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, p. 203–212, Austin, Texas. 38

SUZUKI J., SASAKI Y. & MAEDA E. (2003). Kernels for structured natural language data. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*. 51

TASKAR B., GUESTRIN C. & KOLLER D. (2004). Max-margin markov networks. In *Advances in Neural Information Processing Systems*, vol. 16. 51

THAGARD P., HOLYOAK K.J., NELSON G. & GOCHFELD D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence*, 46(3):259–310. 23

TSOCHANTARIDIS I., HOFMANN T., JOACHIMS T. & ALTUN Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning*, p. 823–830. 51

TURNEY P.D. & LITTMAN M.L. (2005). Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278. 151, 152, 154, 155

TURNEY P.D., LITTMAN M.L., BIGHAM J. & SHNAYDER V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, p. 482–489, Borovets, Bulgaria. 151

VAN DEN BOSCH A. & DAELEMANS W. (1999). Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '99)*, p. 285–292, Maryland, USA. 36, 46

VAN RIJSBERGEN C.J. (1979). *Information Retrieval*. Butterworths, London, UK. 122

VEENSTRA J., VAN DEN BOSCH A., BUCHHOLZ S., DAELEMANS W. & ZAVREL J. (2000). Memory-based word sense disambiguation. *Computers and the Humanities, special issue on Senseval, Word Sense Disambiguation*, 34(1-2):171–177. 36, 46

VOGEL S., NEY H. & TILLMANN C. (1996). Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational linguistics*, p. 836–841. 130

WAGNER R.A. & FISCHER M.J. (1974). The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173. 18, 143, 155

WALLACH H.M. (2004). Conditional random fields: An introduction. Rapport technique MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania. 50

WANG J. & ZUCKER J.D. (2000). Solving the multiple-instance problem: A lazy learning approach. In *Proceedings of the 17thInternational Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA. 30

WESTON J., CHAPELLE O., ELISSEEFF A., SCHÖLKOPF B. & VAPNIK V. (2003). Kernel dependency estimation. *Advances in Neural Information Processing Systems*, 15:873–880. 43, 51

WILSON D.R. & MARTINEZ T.R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286. 58, 127

WOLPERT D.H. & MACREAD W.G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82. 63

YVON F. (1996). *Prononcer par analogie : motivations, formalisations et évaluations*. Thèse de Doctorat, École Nationale Supérieure des Télécommunications, Paris, France. 52, 53, 128, 151

YVON F. (1997). Paradigmatic cascades: a linguistically sound model of pronunciation by analogy. In P.R. Cohen & W. Wahlster (eds.), *Proceedings of the 35thAnnual Meeting of the ACL*, p. 248–435, Madrid, Spain. 52, 53, 151

YVON F. (1999). Pronouncing unknown words using multi-dimensional analogies. In *Proceedings of the European Conference on Speech Application and Technology (Eurospeech)*, vol. 1, p. 199–202, Budapest, Hungary. 102, 151

YVON F. (2003). Finite-state machines solving analogies on words. Rapport technique D008, École Nationale Supérieure des Télécommunications, Paris, France. 2, 75, 78, 79, 80, 116

YVON F., STROPPA N., DELHAY A. & MICLET L. (2004). Solving analogies on words. Rapport technique D005, École Nationale Supérieure des Télécommunications, Paris, France. 78, 80, 104

ZAVREL J., DAELEMANS W. & VEENSTRA J. (1997). Resolving PP attachment ambiguities with memory-based learning. In T.M. Ellison (ed.), *Proceedings of the Conference on Computational Natural Language Learning (CoNLL '97)*, Madrid, Spain. 36, 46

Index des notions

Index des auteurs cités