l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Sofiane MEDJKOUNE

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le label de l'Université de Nantes Angers Le Mans

Discipline : Informatique et applications
Spécialité : Informatique
Laboratoire : Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN)

Soutenue le 13 novembre 2013

École doctorale : 503 (STIM)
Thèse n° : ED 503-206

Stratégies de fusion pour des signaux écrits et sonores

Application à la reconnaissance d'expressions mathématiques

JURY

Rapporteurs :	**M. Thierry ARTIÈRES**, Professeur, Université Pierre et Marie Curie **M. Bertrand COUASNON**, Maître de conférences, HDR, INSA de Rennes
Examinatrice :	**M^me^ Isabelle BLOCH**, Professeur, Telecom Paris Tech
Encadrants :	**M. Harold MOUCHÈRE**, Maître de conférences, Université de Nantes **M. Simon PETITRENAUD**, Maître de conférences, Université du Maine
Directeur de thèse :	**M. Christian VIARD-GAUDIN**, Professeur, Université de Nantes

Remerciements

Ce manuscrit de thèse vient conclure trois années de travaux de recherche effectuées dans le cadre de la préparation de ma thèse. Cet aboutissement n'aurait pas pu être possible sans les conseils, les encouragements et l'aide de beaucoup de personnes. Dans les quelques lignes qui vont suivre je tiens à les en remercier.

Je voudrais commencer par réitérer mes sincères remerciements aux membres du jury d'avoir bien voulu juger mon travail et d'avoir effectué le déplacement. Aux deux rapporteurs, Thierry Artières et Bertrand Coüasnon d'avoir accepté de rapporter sur ma thèse en passant un temps considérable sur sa lecture et son analyse pour me donner des commentaires très pertinents et des suggestions de qualité. A madame l'examinatrice, Isabelle Bloch, qui m'a fait l'honneur de présider ce jury et qui m'a également fait des remarques très intéressantes et très profondes d'un point de vue scientifique.

Je me souviens au tout début de ma thèse, lors de la soutenance d'un des anciens doctorants de mon directeur de thèse, que lorsque ce dernier avait pris la parole, il avait dit que trois ingrédients essentiels sont indispensables pour la bonne marche d'une quelconque thèse. Trouver le bon sujet, trouver le bon candidat et assurer un bon encadrement. En ce qui concerne ce dernier point je ne peux que m'estimer très heureux et je ne remercierai jamais assez mon directeur de thèse Christian Viard-Gaudin, pour sa confiance, sa grande disponibilité et son excellent encadrement mais aussi d'avoir était patient et compréhensif avec moi. Il m'a ouvert grand la porte au monde de la recherche et m'a surtout donné envie d'y rester et de continuer. Mes remerciements s'adressent aussi à mes deux encadrants, Harold Mouchère et Simon Petitrenaud, pour leurs suggestions, leur apport en terme de richesse des idées qu'ils ont apportées aux réflexions menées au cours de la thèse mais aussi la pertinence de leur analyse et toutes les discussions scientifiquement réjouissantes qu'on a échangées.

Mes remerciements s'adressent aussi à l'ensemble des membres de l'équipe IVC sous la direction de Patrick Le Callet, les anciens comme les actuels, qui ont, chacun à sa manière, fait de mes années de thèse des moments très agréables. En particulier : Jonathan Delcourt, Jing Li, Junle Wang, Jinpeng Li, Arlicot Aurore, Bosc Emilie, Zeeshan Ahmad, Cédric Ramassamy et Dalila Goudia, et tous les autres pour leur sympathie et les très bons moments qu'on a passé ensemble, à Florent Autrusseau, Laurent Homa et Romuald Pépion pour leur disponibilité et toute l'aide qu'ils ont pu m'apporter au quotidien. Je remercie également l'ensemble de l'équipe pédagogique du département GEII de l'IUT de Nantes et celle de l'école polytechnique où j'ai eu un grand plaisir à intervenir en enseignement.

Je ne peux pas non plus oublier tous mes nombreux amis pour leur soutien moral et leur disponibilité, mais aussi de me permettre de me changer les idées par moments, ils se reconnaitront...

Je tiens aussi à exprimer ma gratitude à tous les membres de ma famille qui sont ici en France ou ceux qui sont en Algérie, de tous le soutien qui m'ont apporté, j'éviterais de les nommer un par un parce qu'ils sont nombreux (mais surtout pour

éviter d'en oublier certains qui m'en voudront certainement ! !). Ce qui est sûr c'est que je vous en suis tous énormément reconnaissant.

Je ne peux conclure ces quelques ligne de remerciements sans évoquer une personne chère, une personne qui a été là pour moi et m'a apporté son soutien infaillible, en particulier dans les moments de doute. Elle a su être à chaque fois cette bouffée d'oxygène qui me redonne un second souffle pour aller de l'avant. Merci pour tout ma chère Louiza.

A ma petite famille à qui je ne saurais dire assez merci et à qui je dois tout (sans exagérer), mon très cher frère Mohand, mes tendres parents : ma mère Houria et mon père Tayeb, je vous exprime toute ma reconnaissance d'avoir fait de moi ce que je suis aujourd'hui et d'avoir toujours cru (et continuer à croire) en moi. Sans vous trois, je n'aurais certainement pas pu trouver les ressources nécessaires pour aller au bout.

Pour finir, je tenais absolument à dédier ce modeste travail à la mémoire de trois personnes qui me sont chères et qui nous ont quitté au cours de ma thèse et qui n'ont malheureusement pas pu me voir aller au bout de ce projet, eux qui l'espéraient tant : ma grand mère Tassadit, mon cousin Rabah et mon oncle Boudjemaa.

"La philosophie, c'est l'autonomie de la raison. l'homme est né pour être libre, pas pour être soumis à l'arbitraire de l'homme"

Professeur Mohammed Arkoun, historien et philosophe, 1928-2010.

"Quand on est un intello, forcément, on croit au pouvoir des idées... J'étais comme tout le monde... Et comme tout le monde, vous savez ce que c'est ?"

Mouloud Mammeri, écrivain, poète, anthropologue et linguiste, 1917-1989.

Quelqu'un est là mais il n'existe pas (il n'a pas d'impact), un autre existe malgré qu'il ne soit plus là (à travers son œuvre).

Table des matières

Table des figures

Liste des tableaux

Introduction générale

L'ère dans laquelle nous vivons est celle de la communication dans toutes ses variantes. Michel Serres[1], philosophe, historien des sciences et homme de lettres français, l'avait prédit dès les années 1950. Il n'a cessé de soutenir que l'être humain est en phase de vivre la troisième et la plus grande révolution jamais connue dans l'histoire de l'humanité (la première étant l'invention de l'écriture et la deuxième celle de l'imprimerie), et ce grâce à l'avènement du progrès technologique. Le temps lui donne raison et le quotidien des femmes et des hommes actuels est rythmé par le recours à divers dispositifs pour mémoriser, communiquer, apprendre, rechercher, etc.

Faciliter la prise en main de tels dispositifs est de nos jours un objectif principal pour permettre de les rendre les plus accessibles et les plus intuitifs possibles. Historiquement, les êtres humains ont d'abord communiqué principalement par la parole. Par la suite, pour être en mesure de mémoriser et de partager de façon moins contraignante toutes les informations véhiculées par la parole, l'écriture a été inventée. Depuis, ces deux modes d'interaction humain-humain se sont propagés et sont restés les supports naturels et principaux de la transmission de l'information. Cela en fait des moyens privilégiés pour l'interaction humain-machine. En effet, de la même façon que la machine a été créée pour remplir des tâches spécifiques, celle-ci doit pouvoir être en mesure de décoder les signaux de parole et d'écriture auxquels l'être humain s'est déjà familiarisé.

Cette tâche supplémentaire que les machines modernes ont à accomplir est l'objet de recherches intensives. De ces recherches ont émergé des systèmes aboutis capables d'interpréter l'écriture d'un scripteur ou la parole d'un locuteur et d'exécuter les tâches qui y sont associées.

Cette nouvelle génération de systèmes se comporte de façon très efficace quand il est question de langages textuels (parlés et écrits) contrairement à d'autres langages dits graphiques. Ces derniers sont caractérisés par leur structuration spatiale particulière rendant ainsi leur édition par l'écriture ou leur description par la parole plus difficile et plus confuse.

Cette difficulté n'a toutefois pas empêché l'apparition d'applications dédiées à ce type de langage, d'une part pour répondre au besoin évoqué plus haut et d'autre part pour les problématiques très intéressantes que ce langage soulève.

Les expressions mathématiques font partie de ces langages bidimensionnels nécessitant un traitement spécial. L'usage répandu des mathématiques a particulièrement fait du problème de la reconnaissance automatique des expressions mathématiques un des problèmes les plus étudiés dans cette catégorie de langages (quelque 250 publications scientifiques sur le sujet au cours des trente dernières années). Des solutions reposant sur la modalité manuscrite en-ligne ou sur de la parole sont proposées de façon disjointe. Néanmoins, ces solutions se heurtent rapidement à des limitations liées aux propriétés intrinsèques de chacune des deux modalités.

Dans ce travail nous proposons de pallier ces difficultés en explorant une analyse conjointe multimodale des expressions mathématiques. En effet, en considérant

1. http ://www.academie-francaise.fr/les-immortels/michel-serres

les limitations rencontrées par les systèmes mono-modaux existants (écriture manuscrite ou parole) et la forte complémentarité qui existe entre ces deux modalités, nous présentons un système bi-modal qui traite conjointement les signaux de parole et d'écriture manuscrite en-ligne pour la reconnaissance automatique des expressions mathématiques.

L'étude de cette problématique a vu le jour grâce à l'initiative de la région Pays de la Loire qui a émis le souhait de disposer d'un pôle de compétence unique en France qui allierait l'analyse des signaux écrits et sonores pour le traitement automatiques des langues. Cela s'est concrétisé à travers le projet région *DEPART*[2] qui finance les travaux rapportés dans ce manuscrit.

Le système que nous proposons dans ce travail s'inscrit pleinement dans la catégorie des systèmes multi-modaux d'interaction homme-machine. Toutefois, le positionnement qui est le nôtre dans cette thèse est la conception, l'évaluation et l'observation de l'amélioration des performances du système bi-modal relativement au cas d'un traitement mono-modal. On ne s'intéressera pas dans ce rapport à l'étude de l'ergonomie du système proposé. En effet, on ne va pas se préoccuper de la souplesse du système dans sa capacité à s'adapter à l'utilisateur, ni du guidage par rapport à l'état du système et les actions à venir, ni de la concision par rapport au fait de réduire le temps et le nombre de manipulations autant que possible, ni même de l'impact sur le temps de manipulation et la mémorisation des actions... On ne s'intéressera pas non plus au point de vue de l'utilisateur quant à l'impression ressentie face à l'accomplissement de la tâche d'édition des expressions mathématiques dans un cadre bi-modal comparativement au cas mono-modal.

Classiquement, la tâche de mise en œuvre des interfaces homme-machine (IHM) est accomplie en deux phases distinctes qui sont par la suite amenées à être mises en interaction. La première est l'application elle-même qui est en charge de résoudre le problème traité, il s'agit du cœur du système. Dans notre cas, c'est le sujet traité dans cette thèse et correspond à la mise en place de la solution de reconnaissance dans le cadre bi-modal. La seconde tâche quant à elle concerne la réalisation de l'interface à travers laquelle la communication avec l'utilisateur est faite, ce qui est hors du cadre de nos travaux.

Le cadre applicatif et l'aspect bi-modal de l'information considérés dans ce travail de thèse constituent un problème premier en son genre. En effet, nous n'avons recensé aucun travail de ce genre dans la littérature. À partir de là, il était important de disposer de données adéquates au problème, et cela a donc conduit dans un premier temps à effectuer la collecte d'une base d'expressions mathématiques, chacune disponible dans les deux modalité à la fois. Une fois les données à notre disposition nous avons mis en place une architecture modulaire permettant de traiter l'information bi-modale. La modularité de l'architecture proposée permet d'explorer différentes variantes de méthodes de combinaison des deux flux. Les résultats obtenus, même s'ils ne sont pas spectaculaires, montrent une nette amélioration relativement au cas mono-modal basé sur l'écrit seul. Ils montrent également que le

2. http ://www.projet-depart.org/

système proposé tire effectivement profit de la complémentarité écrit-audio.

Le plan de lecture de ce manuscrit est organisé selon deux grandes parties. La première rapporte l'état de l'art de chacune des disciplines impliquées dans cette thèse. La seconde partie quant à elle est dédié aux travaux accomplis au cours de cette thèse. Chacune des deux parties regroupe un ensemble de chapitres. Pour ce qui est de la première partie, elle est composée de quatre chapitres comme suit :

Chapitre 1. Il est consacré à la présentation du langage des mathématiques, au rôle incontournable qu'il occupe dans la vie de tous les jours et aux problématiques scientifiques intéressantes qui peuvent être rencontrées lors de leur traitement automatique à des fins de reconnaissance.

Chapitre 2. Dans ce chapitre, la problématique de reconnaissance des expressions mathématiques manuscrites est abordée. La chaîne de traitement subie par le tracé manuscrit de son état brut jusqu'à l'obtention de l'interprétation de l'expression mathématique est présentée.

Chapitre 3. Ce chapitre revient sur la problématique d'interprétation des expressions mathématiques prononcées. Nous présentons dans ce chapitre le principe des systèmes de reconnaissance de la parole en général et les modifications que ces derniers doivent subir pour s'adapter au module d'interprétation du texte en expression mathématique.

Chapitre 4. C'est un état de l'art sur la notion de fusion qui est rapporté dans ce chapitre. Une présentation des niveaux et des techniques de fusion est faite, suivie par quelques applications recensées dans les travaux de la littérature.

La seconde partie sur les contributions de la thèse est organisée quant à elle en trois chapitres, qui sont les suivants :

Chapitre 5. Dans ce premier chapitre de la partie contribution, la base bimodale *HAMEX* que nous avons mise en place est présentée. Outre la description de la base elle-même, les protocoles de collecte et d'étiquetage manuel sont décrits.

Chapitre 6. Au cours de ce chapitre, dans une première partie nous définissons de façon claire le positionnement de la solution apportée au questionnement soulevé dans cette thèse. Dans la seconde partie de ce chapitre est rapportée la première expérimentation de fusion des flux audio et manuscrit pour la reconnaissance des symboles isolés représentant la première étape de l'étude des expressions complètes. La description de l'architecture du système de reconnaissance utilisé et des méthodes de fusion exploitées est faite ici. C'est aussi dans ce chapitre que sont présentés les premiers résultats de reconnaissance des symboles isolés dans un cadre bi-modal.

Chapitre 7. C'est le dernier chapitre de la thèse. Il est dédié à la présentation du système complet de reconnaissance des expressions mathématiques complètes dans le cadre bi-modal. La description de ces différents modules et des techniques qui y sont mises en œuvre y est faite. Nous analysons également les différents résultats obtenus.

Des conclusions et perspectives de ces travaux viennent clôturer ce manuscrit.

Première partie

État de l'art

Chapitre 1

Un langage graphique incontournable : l'expression mathématique

Sommaire

Le propos de chapitre concerne les mathématiques en tant que langage. L'accent est mis sur le rôle incontournable que celui-ci occupe dans la vie de tous les jours ainsi que les problématiques scientifiques intéressantes qui peuvent être rencontrées lors de son traitement automatique à des fins de reconnaissance.

1.1 Les mathématiques, un langage universel

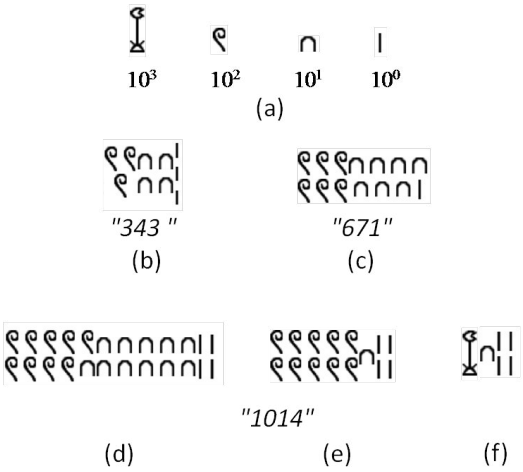
Bien souvent de nombreuses expressions mathématiques sont utilisées au cours d'une thèse pour aider à formaliser des connaissances et permettre un raisonnement rigoureux. Ici, elles serviront avant tout de matériau de base que nous chercherons à transcrire d'une modalité à une autre afin d'en favoriser leur usage indépendamment du domaine scientifique sous-jacent. Commençons par expliciter ce terme *expression mathématique*.

D'un point de vue linguistique, d'après le dictionnaire Larousse 2012, ce groupe nominal est formé d'un nom **expression** qui veut dire littéralement "l'action d'exprimer quelque chose, de le communiquer à autrui par la parole, l'écriture, le geste, la physionomie, etc.". Le second terme est un adjectif : **mathématique**. Ce dernier est défini par le dictionnaire Larousse comme étant "tout ce qui est relatif aux mathématiques. Les mathématiques sont à leur tour la science qui étudie par le moyen du raisonnement déductif les propriétés d'êtres abstraits (nombres, figures géométriques, fonctions, espaces, etc.), ainsi que les relations qui s'établissent entre eux". En résumé, cette composition signifie communiquer et exprimer des choses par le moyen d'un raisonnement connu pour être abstrait. C'est donc un langage de communication universel du fait de son indépendance de la langue usuelle des gens qui y ont recours à travers le monde. C'est aussi un langage compact et très expressif car il permet de traduire des notions diverses et variées, aussi complexes soient elles, par des formulations très condensées sous réserve de savoir les manipuler.

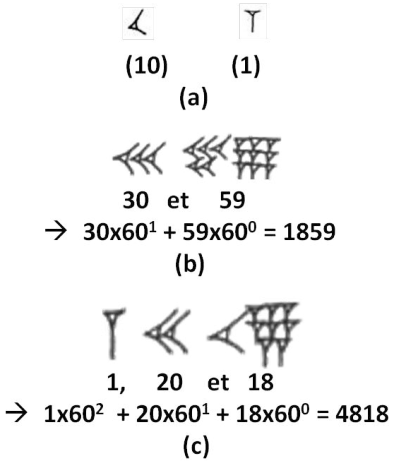
Historiquement parlant, l'usage des expressions mathématiques remonte à plus de vingt-cinq siècles, date à laquelle les premières dénotations de nombres sont apparues [Caj28]. Bien évidement la formulation qui leur est donnée à l'antiquité n'est pas celle qui nous est tout à fait naturelle aujourd'hui. Il en va aussi bien des symboles utilisés que de leur arrangement spatial. Paradoxalement, les mathématiques, cet outil de nature très théorique et manipulant des concepts abstraits permet de résoudre des problèmes pratiques du quotidien. En effet, les premières utilisations de cette science sont apparues dans certaines des civilisations avancées dès l'antiquité. On retrouve des traces de formulation de problèmes de géométrie et d'arithmétique dans les civilisations *égyptienne, babylonienne, chinoise, indienne, etc.* En ce temps là, les gens étaient amenés à répondre à des questions relatives au commerce, au partage de biens et donc à l'évaluation de surface, de périmètre, de volume et à l'exploitation inévitable des opérations de base telles que l'addition/soustraction, la division etc.

On retrouve donc, selon les civilisations, des symboles et des systèmes de numération et de représentation très variés. À titre d'exemple, sur la figure 1.1 sont représentées quelques formules mathématiques des époques égyptienne et babylonienne [Caj28], avec leur transcription en langage mathématique tel que nous le connaissons aujourd'hui.

Sur la figure 1.1.*I*, est représentée une illustration du système de numération égyptien dit additionnel car chaque ordre du nombre est donné par la répétition, autant de fois que nécessaire, du signe associé à cet ordre. Le passage d'un ordre au



I - Le système de numération égyptien : exemple d'une simple addition (en base 10, sans le zéro inconnu alors)



II - Le système de numération babylonien : exemple de représentation des nombres (en base 60)

FIGURE 1.1 – Exemples de systèmes de numération et d'opérations de base dans les civilisations égyptienne et babylonienne [Caj28, TD38]

suivant se fait par rapport à une base décimale amputée du chiffre *zéro*, méconnu à l'époque. Sur la figure 1.1.*Ia*, sont donnés les symboles correspondant au quatre premiers ordres. Les figures 1.1.*Ib* et 1.1.*Ic*, donnent deux exemples de nombres représentés selon ce système. Les figures 1.1.*Id*, 1.1.*Ie* et 1.1.*If*, quant à elles, donnent respectivement le résultat de l'addition des deux nombres précédents sans simplification (report), après report au niveau des dizaines et finalement après report au niveau des centaines (représentation finale du résultat) [Caj28].

La figure 1.1.*II* donne un exemple de représentation des nombres en base 60 selon le système babylonien [TD38], dit positionnel. Seuls deux signes sont disponibles (figure 1.1.*IIa*) : un pour les unités l'autre pour les dizaines. De même que pour le système égyptien, il convient de dupliquer autant de fois que nécessaire un signe pour représenter un nombre entre *un* et *cinquante-neuf*. Il est dit positionnel, car pour exprimer les puissances successives (ordres successifs) de la base il convient de se positionner à gauche de l'ordre inférieur. L'absence d'une représentation vide (*le zéro*), fait que tous les multiples de la base ont la même représentation (c.a.d : 4×60, 4 et 4/60 ont la même représentation par exemple). Les figures 1.1.*IIb* et 1.1.*IIc*, donnent deux exemples de nombre et leur décodage dans notre actuel système décimal de numération.

De nos jours, le langage mathématique a été unifié et est le même où que l'on se trouve dans le monde, le plaçant ainsi au rang de langage universel. Plus que cela, les mathématiques sont devenues incontournables dans la vie de tous les jours. En effet quantité de chiffres, de formules et de diagrammes nous inondent. Comprendre sa facture d'électricité ou le relevé de ses impôts, contracter un prêt voire même bricoler

chez-soi le dimanche, requiert un minimum de connaissance en mathématiques. Bien évidemment, savoir si son meuble passe la porte d'entrée ou estimer la vitesse d'un *quark* ne requiert pas la même connaissance des mathématiques. La seconde question exige des outils mathématiques beaucoup plus sophistiqués et donc l'élaboration d'expressions mathématiques qui peuvent être compliquées.

D'un point de vue structurel, une expression mathématique résulte de la disposition d'un ensemble fini de symboles mathématiques de base dans un plan bidimensionnel. Il est fait état de plus de *250* de ces symboles de base [CWA83]. Il peut s'agir de lettres provenant de différents alphabets (le plus souvent latin ou grec, mais la liste n'est pas limitative), avec leur variantes (type de police, majuscule, minuscule, gras, italique, ...) : $a...z$, $A...Z$, $\alpha...\omega$, $A\ldots\Omega$, $\aleph$, $\beth$ etc., des chiffres arabes, romains ou autres : $0...9, I...M$, etc., des opérateurs binaires : $+, -, \pm, \div, >, <$ etc., des signes de ponctuation : $'$, . ! ? etc., mais aussi d'autres symboles mathématiques, dits élastiques, tels que $\sum$, $\int$, (,] etc. Le tableau 1.1 en fait une synthèse selon les différentes catégories.

Catégorie du symbole	**# Symboles**	**Exemples**
Alphabet latin	52	a...z ; A...Z
Alphabet grec	48	$\alpha...\omega$; $A...\Omega$
Chiffres arabes	10	0...9
Chiffres romains	7	I, V, X, L, C, D, M
Opérateurs élastiques	> 30	$\sum$, $\oint$, $\bigcup$, $\sqrt{}$, $\mapsto$,), etc.
Opérateurs divers (arithmétiques, logiques, relationnels, géométriques ...)	> 100	$+$, $\pm$, $\vee$, $\otimes$, $\exists$, $\in$, $=$, $\equiv$, $\perp$, $\angle$, etc.
Fonctions spéciales	≈ 25	sin, cos, *tanh*, *log*, exp, *lim*, ker, max, etc.

TABLE 1.1 – Les principaux types de symboles courants en langage mathématique

Ces symboles de base sont liés par différents types de relations spatiales (dans un plan 2D) : par rapport à un symbole se trouvant sur la ligne de base, les régions autorisées pour un autre symbole à mettre en relation avec le premier sont définies dans différents travaux [Lav00, DM10] par *en haut à gauche, au dessus, en haut à droite (exposant), à gauche, intérieur (inclus), à droite, en bas à gauche, en bas à droite (indice)*. Dans le tableau 1.2, nous présentons quelques unes des différentes relations possibles ainsi que leurs illustrations à travers des exemples simples.

C'est l'ensemble de ces relations qui donne aux équations mathématiques l'aspect bidimensionnel qui leur vaut d'être étiquetées comme étant un langage graphique.

La composition d'une équation mathématique obéit à des règles de production dépendant d'un contexte particulier. Ce contexte est généralement lié au domaine d'étude abordé. C'est ainsi que les équations mises en œuvre dans le domaine de la physique sont très différentes de celles dont il est question en statistiques, à titre

Dénomination de la relation	Description de la relation	Illustration
Paire horizontale	les symboles sont l'un à la suite de l'autre sur le même axe horizontal	xy
Indice	l'un des deux symboles est en bas à droite de l'autre	x_i
Exposant	l'un des deux symboles est en haut à droite de l'autre	x^2
Au dessus	l'un des deux symboles est positionné exactement au dessus (sur le même axe vertical) de l'autre	$\hat{x}$
En dessous	l'un des deux symboles est sur la même ligne que tout ou partie du corps de l'expression et l'autre symbole est positionné exactement en dessous, sur le même axe vertical, du premier.	$\underrightarrow{x}$
Dedans	l'un des deux symboles est contenu à l'intérieur de l'autre	$\sqrt{x}$

TABLE 1.2 – Les principales relations courantes de base dans le langage mathématique

d'exemple. Une équation peut être valide dans un domaine mais ne pas l'être dans un autre. Pour définir ce qu'est une expression valide, des grammaires sont utilisées. On rapporte dans le tableau 1.3, l'exemple d'une grammaire simple permettant, à partir d'une liste de symboles (dits terminaux) et d'une autre liste de relations autorisées, de générer des expressions mathématiques syntaxiquement et grammaticalement valides.

Règle grammaticale	Type (Description de la relation)
$Sym. \leftarrow x, y, 1, 2, \ldots$	Identité : définit ce qu'est un symbole $Sym.$
$Op. \leftarrow +, -, x, \ldots$	Identité : définit ce qu'est un opérateur $Op.$
$formule \leftarrow SousExp\ Op\ Sym.$	Opérateur : définit une $formule$ comme l'application d'un opérateur au couple ($SousExp$, $Sym.$)
$SousExp \leftarrow Sym.\ Sym.$	Paire horizontale : définit une possibilité de former une $SousExp$ comme l'application de la relation paire horizontale (tableau 1.2, première ligne) au couple ($Sym.$, $Sym.$)
$SousExp \leftarrow Sym.$	Identité : définit une alternative pour former une $SousExp$ comme étant tout simplement un $Sym.$)

TABLE 1.3 – Exemple d'une grammaire simple permettant de générer des expressions mathématiques obéissant à un vocabulaire et à des règles bien précis

Il apparait donc que les expressions mathématiques, indispensables dans bien des domaines, se distinguent d'un texte standard par leurs propriétés géométriques. Dans la section suivante, nous allons aborder cette question plus en détail.

1.2 Spécificités des expressions mathématiques

Le caractère bidimensionnel des expressions mathématiques (EMs) fait que contrairement aux langues classiquement parlées, le recours au support visuel est dans la plupart des cas indispensable pour comprendre de façon non-ambigüe l'EM. Bien souvent, c'est sous forme écrite (sur une feuille de papier, sur un tableau ou grâce à tout autre support) que les EMs sont partagées. Avec l'avènement des ordinateurs, le recours aux documents numériques est de plus en plus prisé. S'il est aussi simple d'éditer un document numérique contenant du texte standard (1D) dans une langue cible donnée via son clavier que d'utiliser un stylo et une feuille de papier, c'est loin d'être le cas pour les EMs. En effet, ces dernières présentent des spécificités que le texte standard ne contient pas [MV98].
Pour faciliter l'insertion des EMs dans les documents numériques, des outils d'édition spécialisés sont mis à disposition. Le degré de complexité de leur usage diffère selon les outils. Cela va des éditeurs très pointus avec des rendus de très bonne qualité mais nécessitant une grande expertise de l'utilisateur (proche des langages de programmation), comme le langage LaTeX dont quelques exemples d'utilisation sont donnés dans le tableau 1.4. De façon complémentaire, il existe les éditeurs de rendus, légèrement moins bons, mais avec une utilisation simplifiée du type *WYSIWYG* (What You See Is What You Get). Ces éditeurs permettent de visualiser immédia-

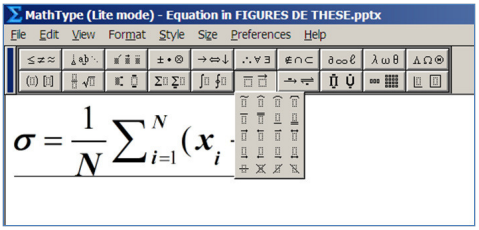
tement l'EM au cours de son édition. Ceci est rendu possible grâce notamment aux divers menus de sélection proposant directement les symboles et les relations sur lesquels il suffit de cliquer pour les inclure dans l'EM en cours d'édition. Quelques uns des outils des plus utilisés de cette deuxième catégorie sont par exemple : *MathType* (figure 1.2a), *MS Equation Editor* de *microsoft* (figure 1.2b), *MathCast* (figure 1.2c) et *MathMagic* (figure 1.2d).

Code LaTeX de l'EM	**L'EM correspondante**
`$ a x^2 + b x + c = 0 $`	$ax^2 + bx + c = 0$
`$ x_{1,2} = \frac{-b\pm \sqrt{b^2 - 4ac}}{2a} $`	$x_{1,2} = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$
`$ f(x)=\int_{-\infty}^{x}{ e^{-t^2}}dt $`	$f(x)=\int_{-\infty}^{x} e^{-t^2}dt$
`$ f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[ a_n \cos \left( nx \right) + b_n \sin \left( nx \right) \right] $`	$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)]$

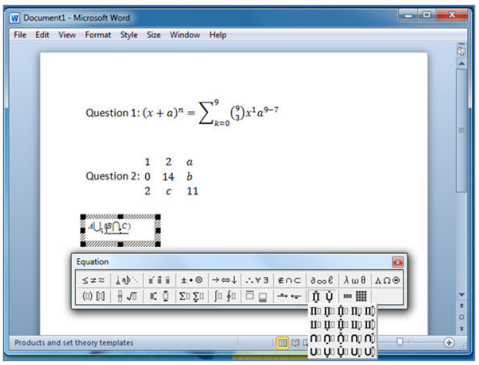
Table 1.4 – Quelques exemples d'expressions mathématiques illustrant la complexité du code LaTeX lorsqu'on augmente la complexité de l'EM (symboles et relations)

L'usage répandu des expressions mathématiques dans des domaines très différents (communautés scientifiques, entreprises, établissements commerciaux et financiers, ...), dont les usagers n'ont pas tous la même expertise, a incité à la recherche de moyens de saisie plus naturels. Ceux-ci font appel à l'interaction gestuelle ou à de la parole, nécessitant de ce fait des systèmes d'interprétation automatiques. À partir de là, les spécificités des EMs introduisent des challenges très intéressants à relever du point de vue scientifique pour ces systèmes en charge de l'interprétation. Ces spécificités, décrites plus haut, sont principalement liées au caractère graphique des EMs, communes à tous les langages de ce type, mais pas seulement. En effet, certaines propriétés sont propres au langage mathématique. Globalement, on identifie deux grandes catégories de causes d'ambiguïtés faisant des EMs une exception :

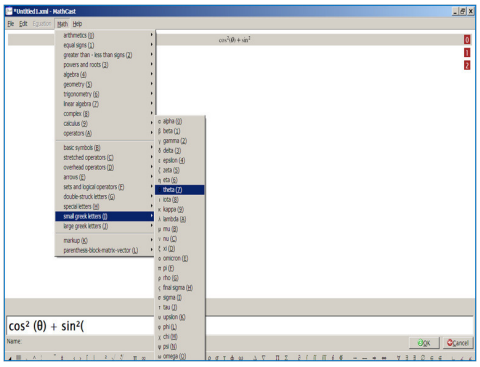
1) Le nombre très élevé de symboles et leur variété (tableau 1.1).
2) La possibilité de dispositions complexes de ces symboles les uns relativement aux autres, comme par exemple l'application récurrente des exemples de relations de base données dans tableau 1.2 (indice d'un exposant, fraction en exposant, ...).



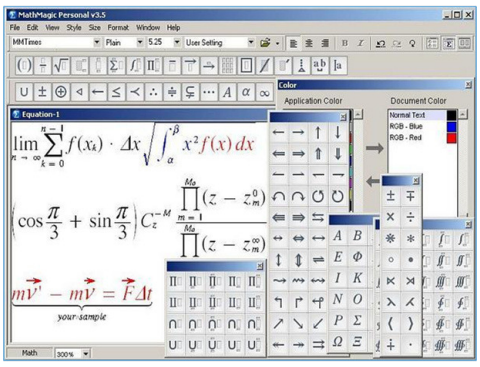
(a) Éditeur *MathType*



(b) Éditeur des EM de *Microsoft*



(c) Éditeur *MathCast*



(d) Éditeur *MathMagic*

FIGURE 1.2 – Exemples d'éditeurs du type *WYSIWYG* les plus utilisés.

1.2.1 Ambiguïtés engendrées par le nombre élevé de symboles et par leur variété

Comme il a été spécifié dans le tableau 1.1, le nombre de symboles impliqués dans le langage mathématique est très important (supérieur à 250). Ceci n'est pas sans conséquence quant à la réponse à apporter concernant leur classification par un moyen automatique, voire même humain. En effet, que l'on considère leur dictée ou leur tracé manuscrit par exemple, certaines classes de symboles sont difficiles à distinguer. Les raisons de ces **confusions inter-classes** sont variées. Dans la suite nous en faisons un résumé.

- **Le caractère minuscule ou majuscule de certaines lettres de l'alphabet** rend parfois difficile l'association de la bonne étiquette au symbole dans le cas d'un signal manuscrit. C'est notamment le cas des lettres "c" et "o", tel qu'illustré sur la figure 1.3. Cette difficulté, si elle est surmontable dans le cas d'un texte standard grâce au contexte, puisqu'une majuscule est présente en début de phrase et des noms propres, dans le cas des EMs ce n'est pas du tout le cas. En effet dans une EM, une majuscule peut se trouver dans

n'importe laquelle des positions. Par contre, en considérant le contexte global de l'EM, en regardant les symboles voisins, cette ambiguïté peut être réduite. Dans le cas de la modalité audio, cette ambiguïté peut exister ou non. En effet, dans le cas où la dictée est rigoureuse en précisant la casse de façon explicite, comme par exemple "petit c", "c majuscule", ..., la confusion n'existe pas. En revanche, lorsque cette précision est omise, l'ambiguïté est totale.

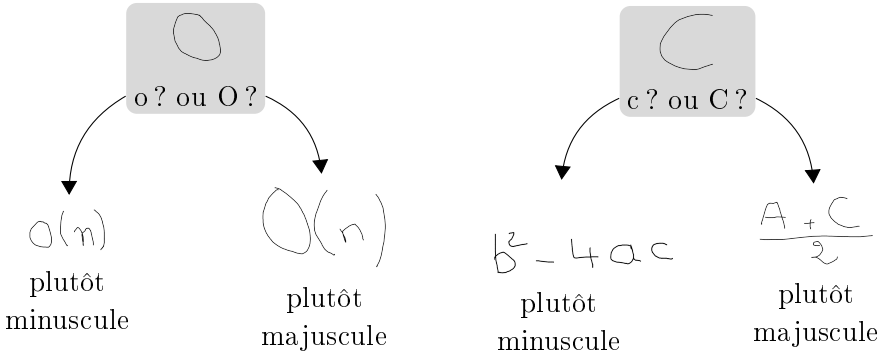


FIGURE 1.3 – Exemples d'ambiguïté liée à la casse (majuscule ou minuscule) des lettres : cas de o et de c

- **Les formes géométriques de certains symboles sont très proches.** En effet, à l'image des exemples rapportés en figure 1.4 concernant les symboles "q" et "9" ou encore "(" et "c" , même s'ils sont sémantiquement différents, les tracés de certains symboles présentent des similitudes prononcées. Afin d'améliorer la robustesse des systèmes de reconnaissance automatique faisant face à ce type de problèmes, il est indispensable de disposer de suffisamment d'exemples de chacune des deux classes en conflit (avec des styles d'écriture très variables pour couvrir la variabilité intra-classe et mieux définir les frontières inter-classes). Cela permet de définir au mieux les frontières entre elles. Par contre, cela ne garantit pas de s'affranchir de façon définitive de cette difficulté.

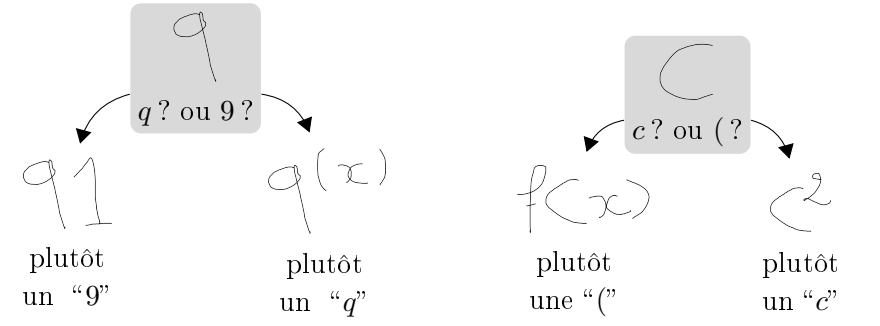


FIGURE 1.4 – Exemple d'ambiguïté liée à la similarité des formes géométriques de certaines classes de symboles manuscrits

- **Les formes acoustiques de certains symboles sont très proches.** Du point de vue de la parole, il existe des symboles dont la prononciation est très similaire et fait intervenir quasiment la même séquence phonétique. Ceci est d'autant plus avéré si l'articulation du locuteur n'est pas des plus fiables. C'est le cas par exemple des symboles "a" et "1" ou dans une moindre mesure de "u" et "μ". Dans d'autres exemples, un symbole peut être confondu avec une séquence de symboles, comme c'est le cas pour la lettre grecque "κ", prononcée *kappa*, et la concaténation des lettres "k", "p" et "a". Dans ce cas aussi, face à ce genre de confusions, la qualité de la prononciation a un rôle déterminant dans la solution que proposerait un système de transcription automatique d'un tel signal.
- **Le rôle de certains symboles change avec le contexte.** L'équation représentée sur la figure 1.5, montre l'exemple d'une même forme représentant des symboles ayant des étiquettes différentes. Les trois traits en vert pris de façon isolée, correspondent tous au signe d'une barre horizontale. Il est impossible de leur associer une étiquette valide sans savoir le rôle qu'ils tiennent au sein d'une EM. En effet, comme le montre cet exemple, la plus grande de toutes est une barre de fraction (d'un fait qu'elle contienne des sous expressions en dessous et au dessus). Au sein de la sous expression se trouvant au numérateur, la barre horizontale est cette fois-ci associée à un signe de l'opération de soustraction. La troisième barre verte, quant à elle, est associé avec une autre dans une disposition d'alignement vertical, pour former le signe d'égalité.

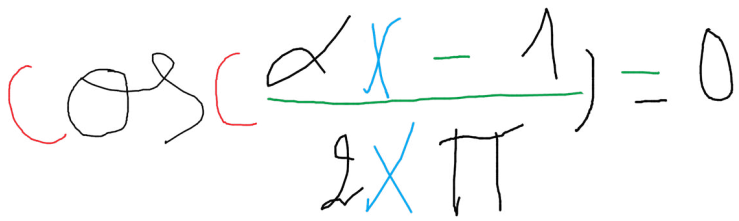


FIGURE 1.5 – Exemple d'expression mathématique présentant des ambiguïtés qui ne peuvent être levées sans le contexte.

- **Certains symboles peuvent être des parties d'autres symboles.** Des exemples de ce type d'ambiguïté sont également représentés au sein de la figure 1.5. Dans ce contexte, un tracé pouvant représenter un symbole donné se trouve être un morceau d'un symbole plus grand. C'est le cas du premier trait rouge de la figure 1.5 pouvant se voir attribuer l'étiquette "c", mais c'est sans compter sur le reste du symbole suggérant qu'il soit un élément du symbole "cos". Dans le signe d'égalité ("=") qui est formé par l'association de deux traits horizontaux élémentaires, ceux-ci peuvent chacun être associés à un autre signe, le "$-$".

En plus de tous les points, liés au grand nombre de symboles, présentés dans cette section, la variabilité des tracés associés à un symbole et les différents styles des scripteurs ou des locuteurs ne font que renforcer la variance *intra-classes* et réduire

(a) Espace de recherche total (9 dispositions différentes au total)

$$abc \quad acb$$
$$bac \quad bca$$
$$cab \quad cba$$

(b) Partie de l'ensemble de toutes les dispositions possibles (72 au total)

$$a^{b^c} \quad a^{bc} \quad a^{b_c} \quad a^b_c$$
$$a^b c \quad ab^c \quad abc \quad ab_c$$
$$a^c_b \quad a_{bc} \quad a_{b_c} \quad a_b c$$

FIGURE 1.6 – Exemple montrant l'étendue de l'espace de recherche des relations possibles entre trois symboles a, b et c dans le cas 1.6a) du texte standard et 1.6b) d'une expression mathématique [RK09]

l'écart *inter-classes*. De plus, certains symboles sont connus pour être très facilement assimilables à du bruit. C'est le cas des petits symboles tel que les points "." ou les virgules "," dans le cas du signal manuscrit. C'est également le cas des symboles contenant des *fricatives* dites *sourdes* : *[f], [s] et [ʃ]*, qui sont facilement confondues à du bruit de fond dans le cas de la modalité parlée.

1.2.2 Ambiguïtés liées au caractère 2D des EMs

Pour interpréter une EM, il ne suffit pas seulement d'identifier tous les symboles qui la composent. Il reste une autre tâche qui peut se montrer encore plus ardue. Il s'agit de retrouver le modèle spatial selon lequel ces symboles de base s'arrangent. De façon similaire au cas des symboles, nous présentons dans ce qui suit les problématiques majeures que soulève l'analyse structurelle des EMs.

- **L'espace de recherche est très vaste.** Avant toute chose, le caractère 2D des EMs, autorisant des possibilités de disposition suivant les différentes directions, fait que l'espace des hypothèses de relation est nettement plus élargi, comparé à celui d'un texte standard (*cf.* figure 1.6). Combinée avec la variété et le grand nombre de symboles, cette propriété est même plus contraignante dans le cas des EMs comparativement aux autres langages graphiques tels que les *équations chimiques* [ZSY09], les *circuits électriques* [FVGS09], les *diagrammes* [AFMVG11], les *partitions musicales* [RCC10], ou même les *plans d'architecture* [Gho12] où le nombre de symboles est nettement plus réduit.
- **Les frontières séparant les différentes relations spatiales sont de nature floue.** Les relations présentées dans le tableau 1.2 semblent bien nettes, du moins du point de vue théorique. Cela est loin d'être le cas dès qu'il s'agit de se positionner du point de vue pratique. Si différentes personnes écrivaient sur une feuille de papier une EM contenant une simple relation d'exposant (x^2 à titre d'exemple), on obtiendrait une grande variabilité dans le positionnement du "2" par rapport au "x". Cette variabilité traduit la liberté dans le geste ressentie par les scripteurs. Le contexte peut aussi avoir une influence : une équation contenant uniquement cette relation, ou au contraire qui serait une partie d'une EM globale ne mèneraient pas au même positionnement. La

liberté que s'autorise le scripteur lorsqu'il effectue son geste, dans cet exemple, traduit très bien l'ambiguïté de frontière entre les relations *complètement en haut, complètement à droite et en haut à droite* comme l'illustre l'exemple de la figure 1.7. Dans cet exemple, on montre les différentes positions spatiales possibles relativement à un symbole de base (ici le x). Plus la zone est claire moins la zone est conflictuelle. À l'inverse, plus la zone est sombre, plus le conflit est grand.

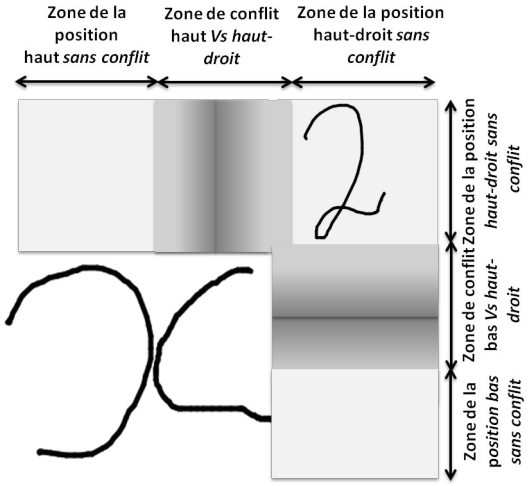


FIGURE 1.7 – Exemples de conflits pouvant exister entre les positions relatives des symboles : exemple des positions haut, haut-droit et droit. Plus la zone est sombre, plus le conflit est grand.

Autrement dit, sur l'ensemble des positionnements du symbole "2" dans tout l'espace 2D autour de "x" , dans cet exemple, il existe des positions où on peut se prononcer de façon sûre et unique sur la nature de la relation (dessus, exposant, paire horizontale, indice, ...) et des zones d'ambiguïtés importantes. Sur la figure 1.8 on a indiqué des exemples d'interprétation des configurations de position des symboles "x" et "2" selon la représentation faite dans l'exemple de la figure 1.7 pour les deux positions : *exposant, paire horizontale*. Encore une fois, si les cas extrêmes sont bien définis, la relation est graduellement de plus en plus floue entre ces deux situations et conduit même à un cas d'ambiguïté maximale pour lequel une prise de décision est quasiment aléatoire.

- **L'interprétation des relations dépend du contexte.** Ainsi que l'a noté Martin dans [Mar71], résoudre l'ambiguïté présentée précédemment ne suffit pas pour déduire de façon certaine la relation. En effet, il faut composer avec le positionnement relatif des symboles. Ceci est d'autant plus vrai, quand on sait que la situation est plus complexe dès que plus de deux symboles sont impliqués. À l'image de l'exemple donné sur la figure 1.9, avec simplement trois symboles, on se retrouve dans une situation assez délicate quant à l'in-

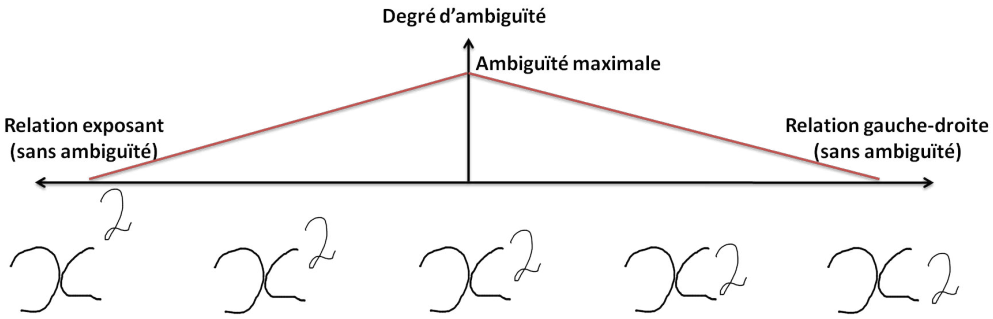


FIGURE 1.8 – Exemple d'ambiguïté au niveau relations : cas de la transition *exposant*$\longleftrightarrow$*paire horizontale*.

terprétation finale à donner à l'expression, si le contexte global n'est pas pris en compte. En effet, sans le contexte, on serait tenté de se prononcer en faveur d'un relation d'indice (*z est en indice de y*). Ce résultat serait valide dans le cas de l'EM de droite (xy_z), mais tout à fait erroné dans le cas de celle de gauche $x^y z$. D'où l'importance du contexte global durant la phase d'interprétation.

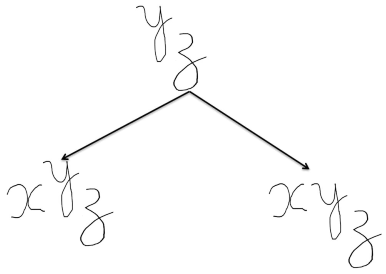


FIGURE 1.9 – Exemple d'ambiguïté induite par le positionnement relatif des symboles

- **La description orale (naturelle) des relations est parfois très vague et peu discriminante.** Du point de vue de la parole spontanée, il est assez souvent très difficile de prévaloir certaines relations au profit d'autres quand une description textuelle est disponible. En effet, si des règles de diction ne sont pas imposées, un texte décrivant une EM peut avoir différentes écritures en langage mathématique [Fat98]. Une illustration de cela est rapportée sur la figure 1.10. Dans cet exemple, la description de ce qu'on peut appeler une EM très simple : *"a plus b sur c plus d "*, laisse énormément de libertés d'interprétation possibles.

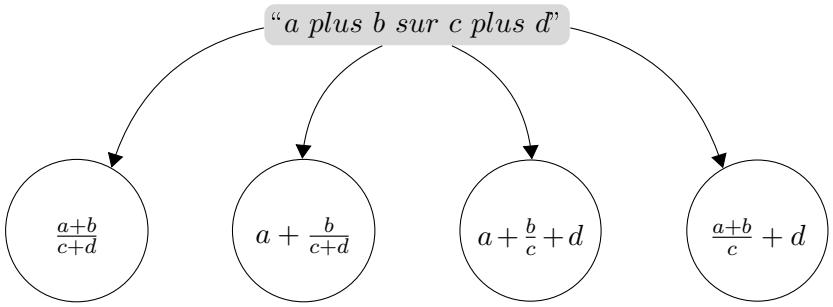


FIGURE 1.10 – Exemple d'ambiguïté liée à la diction de l'EM : différentes interprétations possibles de la description textuelle

1.3 Bilan

Ce chapitre a pour vocation d'argumenter le double intérêt à la fois pratique et scientifique du sujet abordé dans cette thèse, à savoir les **expressions mathématiques**.

Dans la première partie, après avoir défini de façon précise ce qu'est une EM suivant différents points de vue (linguistique, historique et structurel), nous avons mis en exergue l'importance des mathématiques dans de nombreux domaines. Nous sommes également arrivés à la conclusion de la nécessité de moyens automatiques de gestion (saisie, modification, recherche d'information, ...) de telles structures.

La seconde partie quant à elle, est dédiée à l'aspect structurel qui caractérise l'EM. L'extraction automatique de cette structure sera une problématique majeure d'un système de reconnaissance automatique des EMs.

En conclusion, le besoin d'outils de gestion de contenu mathématique numérique qui seraient plus intuitifs et plus adaptés, combiné avec les possibilités qu'offre le progrès technologique des systèmes d'interaction homme-document justifient l'intérêt de l'étude d'un système s'appuyant sur un signal manuscrit en-ligne et/ou un signal de parole. Conséquemment, on note un intérêt de la communauté scientifique, surtout dans le domaine manuscrit en-ligne pour proposer des approches originales cherchant à répondre aux défis spécifiques posés par les EMs.

Dans les chapitres 2 et 3, nous reviendrons sur ces nouveaux modes basés sur les flux écrit et sonore pour l'édition de documents numériques contenant des EMs.

Chapitre 2

Reconnaissance des expressions mathématiques manuscrites

Sommaire

Ce chapitre est dédié à la problématique de reconnaissance des expressions mathématiques manuscrites. La chaîne de traitement conduisant à l'interprétation du tracé manuscrit de son état brut jusqu'à l'obtention de l'interprétation de l'expression mathématique sera présentée.

2.1 Introduction

Se servir de son stylo pour décrire une expression mathématique semble être le moyen le plus naturel, il est le premier mode avec lequel on a appris à manipuler ce genre de langage en étant jeune à l'école. Ceci est d'autant plus vrai que le caractère 2D des EMs est directement perceptible à partir du signal manuscrit (l'image résultante). Il va de soit que le partage d'information disponible sous forme d'une expression mathématique entre les humains se prête très bien à cette forme de représentation de la connaissance (écriture manuscrite). Un enseignant aura toujours le réflexe de s'appuyer sur un support visuel (tableau classique ou numérique, diapositives, ...) pour présenter les EMs de son cours à ses étudiants.

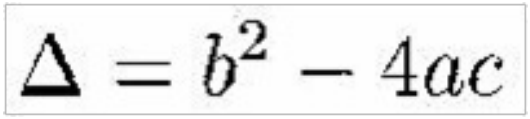
En revanche, insérer des EMs dans un document numérique, à partir d'un tracé n'est pas trivial. En effet, en plus de l'interface matérielle nécessaire pour converser avec sa machine, une interface logicielle est requise pour assurer la traduction du langage de l'humain (l'écriture dans ce cas) vers celui de son ordinateur.

Le progrès technologique a apporté (et continue d'apporter) des solutions très efficaces quant à la première interrogation, concernant l'aspect matériel. Ceci en proposant des outils de capture du signal manuscrit performants et variés. Ils peuvent être de deux types : le premier consiste à prendre une photographie de la structure représentant la formule mathématique, on parle alors de signal **hors-ligne** (dit **off-line** en anglais). Le second type d'outils fournit une information dynamique du tracé : la trajectoire suivie par le stylo décrivant l'EM est obtenue sous forme de séquences de points. Au niveau de chaque point plusieurs types de renseignements peuvent être disponibles : en plus des coordonnées (x, y), on retrouve généralement des informations de temps et de pression. On parle dans ce second cas de signal **en-ligne** (dont la dénotation anglo-saxonne est : **on-line**).

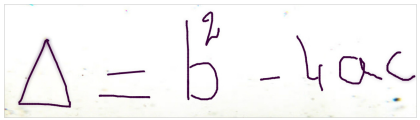
L'interface logicielle de son côté, ne s'est pas faite attendre. En effet, dès l'apparition de la première interface graphique de communication homme-machine vers 1964, la *tablet RAND* [DE64], l'un des premiers systèmes de reconnaissance d'expressions mathématiques a vu le jour [And67] trois ans plus tard. Depuis, les systèmes n'ont cessé de progresser et de gagner en robustesse et en généricité. Ceci est d'autant plus vrai avec le bond technologique fait au niveau matériel en proposant des interfaces d'interaction toujours plus fiables et plus performantes fournissant une information d'une précision croissante [KASA08]. En fait, nous pouvons identifier deux grandes classes de systèmes de reconnaissance des expressions mathématiques et ceci en considérant la nature du signal qui y est manipulé : *les systèmes hors-ligne.* et *les systèmes en-ligne [CY00, TR07].* Dans le cas des systèmes hors-lignes, il est également possible d'identifier deux catégories, basées sur des images : de *l'écriture manuscrite* [KWL95, CY00], ou de *l'imprimé* [FTBM96, UNS05].

Bien que la nature du signal soit différente dans chacune des catégories susmentionnées, reconnaitre une expression mathématique s'opère suivant deux niveaux, abstraction faite de la nature du signal en entrée (en/hors-ligne) [BG97, ZB11]. Ces deux niveaux sont nommément : l'**extraction des symboles** faisant partie de l'EM et la **découverte des relations spatiales** les liant les uns aux

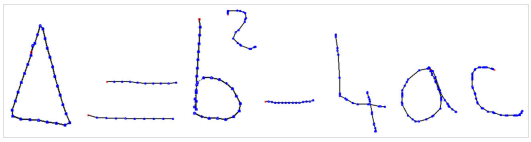
autres. Bien évidemment, selon le type de signal, les deux niveaux identifiés doivent être adaptés. Par exemple, lorsque l'EM à reconnaître est présentée sous forme d'une image (tracé hors-ligne), un symbole est défini comme une forme composée de pixels noirs (au sein d'une image binaire à fond blanc), qui peuvent éventuellement ne pas être connexes. Si c'est la version en-ligne de l'EM qui est à analyser, le symbole dans ce cas est la séquence de points représentant la trajectoire du stylo lors de son écriture. Cette séquence de points peut être composée d'une ou plusieurs sous-séquences. Une sous-séquence est définie par deux points extrêmes : *le poser* et *le lever* de stylo (dont les équivalents anglo-saxons sont respectivement *PenDown* et *PenUp*). Ces deux points renseignent sur les instants où l'on pose le stylo sur la feuille pour entamer le tracé et le moment où on a fini d'écrire la forme en question et où le contact stylo-feuille est rompu. C'est cette sous-séquence délimitée par ces deux points remarquables qui est l'unité de base nommée **trait** (plus communément citée par son équivalent anglophone **stroke**). Sur la figure 2.1 est montré un exemple d'expression mathématique suivant les trois types de formes définies précédemment.



(a) EM imprimée hors-ligne



(b) EM manuscrite hors-ligne



(c) EM manuscrite en-ligne

FIGURE 2.1 – Exemples d'une expression mathématique disponible sous les trois formats.

Nous donnons sur le schéma de la figure 2.2 un récapitulatif succinct de l'enchaînement des deux processus nécessaires à la reconnaissance d'une EM (identification des symboles et analyse structurelle) et ce pour un signal d'entrée de type quelconque.

Les différentes phases de reconnaissance des EMs sont comme suit :

- La segmentation pour construire les hypothèses de symboles.
- La reconnaissance pour associer des étiquettes aux segmentations identifiées.
- L'analyse structurelle qui vise à construire l'arbre donnant l'organisation spatiale des symboles identifiés.
- La validation syntaxique pour vérifier la validité de l'interprétation finale par la grammaire pilotant la reconnaissance.

Ces étapes ne sont pas forcément accomplies d'une façon séquentielle. En effet, différentes stratégies ont été proposées décrivant la façon de faire interagir ses modules,

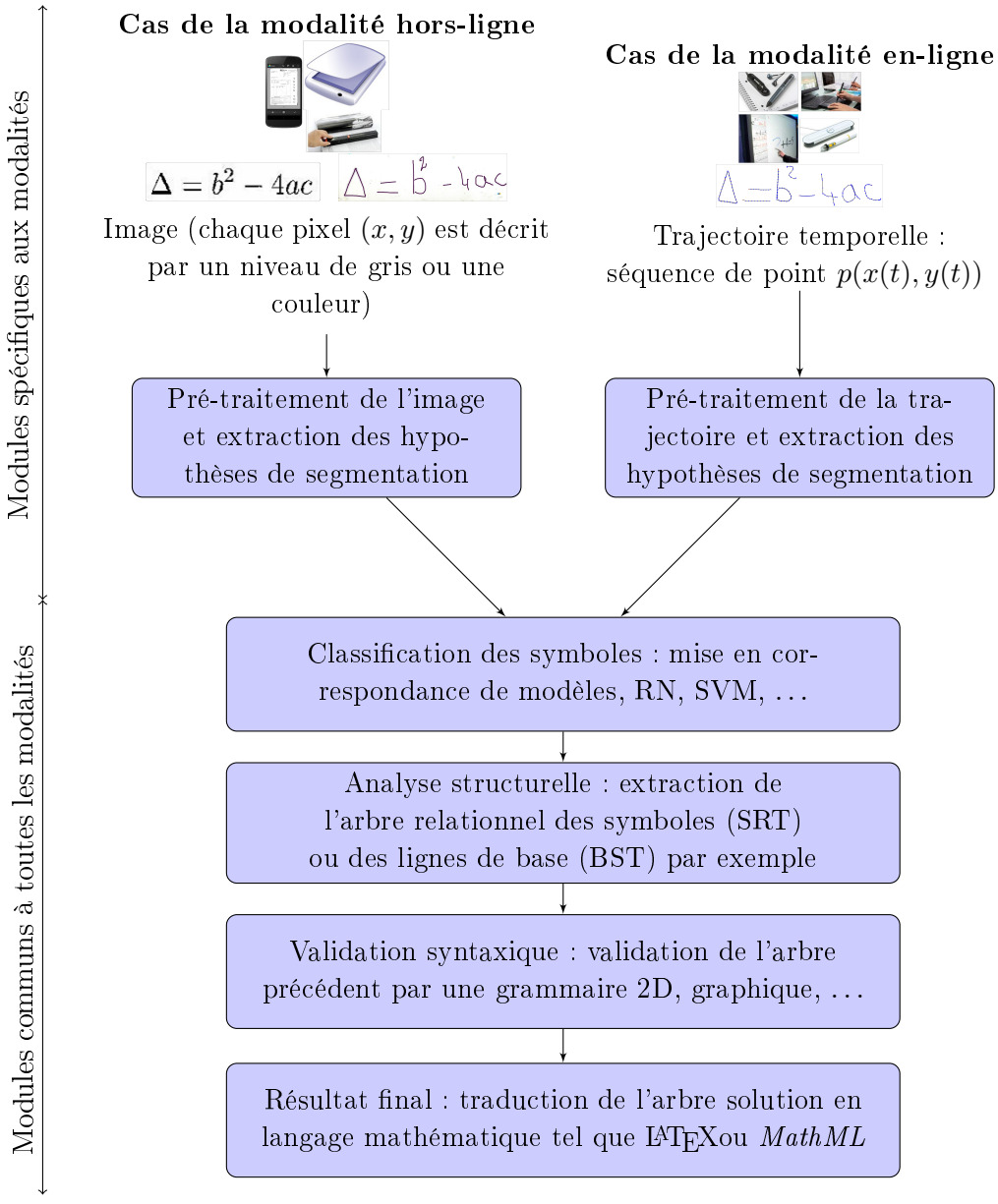


FIGURE 2.2 – Étapes principales du processus de reconnaissance automatique d'une EM à partir de son tracé manuscrit

nous en reparlerons dans la suite de ce chapitre. On représente sur la figure 2.2 ces différentes étapes pour les deux types de signaux en-ligne et hors-ligne.

Avant d'aller plus loin dans la description des différents modules et de la façon de les associer, nous prenons partie pour ce qui concerne la modalité manuscrite de nous focaliser sur le tracé en-ligne. En effet, dans notre travail, il est essentiellement question des EMs en-ligne. De ce fait, dans la suite, par abus de langage et sauf

mention contraire, on désignera par *signal (ou expression) manuscrit(e)* sa version en-ligne. D'ailleurs, mises à part les particularités de pré-traitements spécifiques à chacune des formes du signal, les processus suivants restent identiques (*cf.* figure 2.2).

À présent, nous passons à la description d'un système type de reconnaissance d'expressions mathématiques en-ligne. Nous discuterons non seulement du détail de chacun de ces modules en référence à la littérature, mais également de l'interaction inter-modules pour analyser le type d'information commune entre eux.

2.2 Retour sur le signal manuscrit en-ligne, qu'est-ce qu'un symbole ?

Pour développer ce qui a été présenté en introduction, dans le cas du signal manuscrit en-ligne, le système de reconnaissance automatique reçoit en entrée un ensemble de traits élémentaires, les **strokes**, décrites sous forme de séquences de coordonnées délimitées par deux points remarquables : **le poser** et **le lever** du stylo. En plus de l'information de position (x, y), pour chaque point du trait, le signal manuscrit embarque naturellement l'information de temps et quelquefois celle de la pression du stylo. Cette richesse d'information, comparée au signal hors-ligne, permet de retrouver exactement l'historique du tracé. Cette technologie permet de composer un symbole à l'aide d'un seul trait élémentaire ou bien en s'autorisant plusieurs levers/posers de stylo (symbole formé de plus d'un trait). Néanmoins, dans l'état de l'art, on suppose en général que deux symboles distincts ne peuvent partager un même trait. Cette "contrainte" durant la saisie semble assez naturelle et peu contraignante. Il en résulte qu'il n'est pas nécessaire de segmenter un trait élémentaire. Pour autant, l'étape de segmentation est loin d'être triviale. En effet, il reste indispensable de regrouper correctement l'ensemble des traits appartenant à un même symbole. A ce stade, deux hypothèses sont rencontrées. Certains systèmes imposent une séquence stricte pour former la partition correspondant à la segmentation. Dans ce cas, il est obligatoire de terminer un symbole avant de commencer le suivant. D'autres systèmes, notamment celui que nous mettrons en œuvre, relâchent cette contrainte et permettent de revenir sur un symbole déjà commencé. Ainsi, il est possible de corriger ou compléter l'expression à tout moment. Par exemple, après avoir écrit l'expression "$a = b$", il est possible de revenir sur le symbole '=', de le compléter pour le transformer en '$\neq$' pour obtenir l'expression "$a \neq b$".

La question de la segmentation est donc cruciale. Elle conditionne bien sûr la suite des traitements. Généralement, il ne s'agit pas de produire une seule segmentation mais un ensemble d'hypothèses de symboles. C'est ce point qui est développé dans la section qui suit.

2.3 Segmentation des expressions mathématiques

Le but de cette étape est de regrouper les traits élémentaires de sorte à former une partition de l'ensemble des traits. Parmi les hypothèses de symboles qui vont

être formées, certaines vont correspondre à des hypothèses valides : tous les traits d'un symbole sont présents, d'autres vont correspondre à des hypothèses erronées. Il peut s'agir alors de situations de sur-segmentation où seule une partie du symbole est présente, ou à l'inverse d'une situation de sous-segmentation lorsque des traits de plusieurs symboles sont regroupés. Une tâche subséquente sera d'attribuer une étiquette, une classe de symbole, à chacune des hypothèses de symboles. Ce sera alors le rôle du classifieur de symboles qui devra alors avoir la capacité de rejeter les hypothèses erronées. Ce point sera abordé un peu plus tard dans la section 2.4. Dans le cas idéal, en consommant tous les traits élémentaires, chacun une fois au maximum (hypothèse d'appartenance de chaque stroke à un et seulement un seul symbole), on aura retrouvé tous les sous-ensembles de traits formant effectivement des symboles. Le caractère non séquentiel, énoncé dans la section 2.2, rend le recours à l'aspect temporel du signal manuscrit pour accomplir la tâche de segmentation peu fiable.

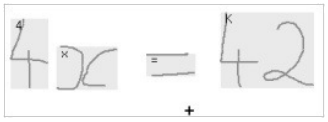
Dans le cas du signal en-ligne, on rencontre dans la littérature deux catégories de segmentation. La première se déroule au cours de l'acquisition du signal (à la volée) et n'est donc envisageable que dans le cas du signal en-ligne. Dans ce cas le contexte est très local et se limite aux traits déjà tracés à un instant t donné [SNA99, VHH08]. Ce mode opératoire est très adapté aux interfaces interactives de saisie car il permet à l'utilisateur de détecter et corriger immédiatement, si possible, les erreurs. C'est ce que proposent Smithies et al. dans [SNA99], où toutes les hypothèses de segmentation contenant au maximum quatre strokes sont formulées et envoyées au classifieur. L'hypothèse dont le score est le plus fort est retenue et les traits qui la composent ne seront plus considérés par la suite. Les autres strokes quant à eux sont remis en jeux et vont être complétés par d'autres en cours de saisie. Une fois que quatre autres traits sont disponibles, le processus recommence jusqu'à la fin de la saisie. Dans leur système, Smithies et al. autorisent également la correction d'éventuelles mauvaises segmentations en sélectionnant manuellement les traits devant former un symbole et les éventuelles mauvaises classifications des bonnes segmentations en déroulant une liste des N meilleures reconnaissances. La figure 2.3 donne une illustration de cette procédure. Sans les possibilités de correction, ce type de systèmes souffre du fait que la validation d'une segmentation repose grandement sur la qualité du classifieur servant à faire le tri entre les segmentations proposées pour un jeu de traits élémentaires donné.

Dans ses travaux de thèse, Ernesto Tapia [TR03, Tap04], propose également une segmentation à la volée. Dans son cas, chaque nouveau stroke qui vient d'être saisi ne va être regroupé au sein de la l'hypothèse de symboles précédente que s'il est suffisamment proche, sinon il sera considéré comme faisant partie d'une nouvelle hypothèse à construire. L'inconvénient majeur est que cela nécessite de définir un seuil sur la distance à partir duquel on admet qu'il faille créer une nouvelle hypothèse de symboles.

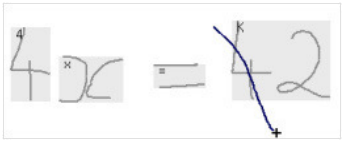
La seconde approche de la segmentation suppose la présence de la totalité des traits composant l'EM au moment de son accomplissement. C'est le cas de l'immense

(a) Segmentation initiale (5 symboles)

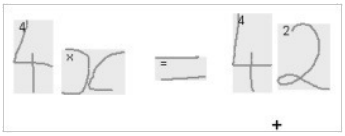
(b) Détection d'une sur-segmentation par l'utilisateur et sollicitation du système pour une correction



(c) Résultat courant après correction de la sur-segmentation détectée (4 symboles)



(d) L'utilisateur détecte une sous-segmentation et demande la correction au système



(e) Résultat final valide (5 symboles)

FIGURE 2.3 – Mode opératoire du système de reconnaissance de Smithies [SNA99]

majorité des systèmes mentionnés dans la littérature. En procédant de la sorte, le contexte global de l'EM est disponible et est susceptible de guider la segmentation pour accroître sa précision.

Identifier le découpage optimal de l'ensemble des strokes en symboles peut être achevé selon deux points de vue.

Le premier, historique, vient de la modalité hors-ligne et ne prend en compte que les informations spatiales du signal : les coordonnées (x, y). Des projections suivant les axes x et y s'en suivent, pour que les creux des profils soient utilisés comme points de segmentation. Des raffinement sont apportés pour prendre en compte la particularité de certain symboles tels que : `$i$´, `j´ ou encore `$=$´. Des symboles du même type que la racine carrée `$\sqrt{\ }$´ bénéficient d'un traitement particulier du fait qu'ils peuvent contenir en leur intérieur d'autres symboles. Pour plus de détails à ce sujet, on peut se référer aux travaux de Okamoto et al. [OM92]. Dans la même catégorie de méthodes (partitionnement dans l'espace 2D), on retrouve les travaux de Ha et al. [HHP95] qui ont exploité les boîtes englobantes de primitives extraites de l'image d'une EM imprimée au lieu de l'information extraite des pixels. Ces primitives (susceptibles d'être combinées) sont ensuite utilisées pour construire une certaine hiérarchie de tous les objets (segments portés par les boîtes englobantes). Une application de cette approche dans le cas du signal en-ligne est celle présentée par Chan et al. dans [CY01].

Le second point de vue a vu le jour avec l'arrivée du signal en-ligne et tire profit

de la richesse de ce type de signaux. En admettant qu'un découpage optimal et unique de l'ensemble des traits en symboles existe, il s'agit de faire le tri parmi tous les partitionnements possibles et d'en choisir le plus vraisemblable et le moins coûteux. La taille $N_{seg}(N_{strk})$ de l'espace de recherche (entendre par là, le nombre de partitions possibles) dans ce cas est directement liée au nombre de traits N_{strk} composant l'EM. Il est donné par le nombre de Bell [1] rapporté par l'équation 2.1. Ce nombre croît très rapidement avec le nombre total de strokes. Pour réduire cet espace, des contraintes de proximité géométrique et de nombre maximal de traits composant un symbole sont ajoutées. Même en opérant de la sorte, énormément de mauvaises segmentations sont générées.

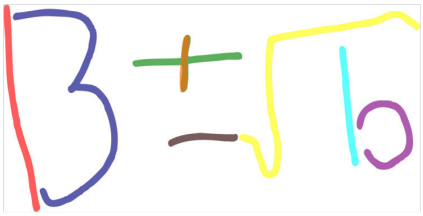
$$\begin{cases} N_{seg}(N_{strk}) = \sum_{k=0}^{N_{strk}-1} \binom{N_{strk}}{k} N_{seg}(k) \\ N_{seg}(0) = 1 \\ \binom{N_{strk}}{k} = \frac{N_{strk}!}{(N_{strk}-k)!k!} \end{cases} \tag{2.1}$$

Sur la figure 2.4 sont données quelques partitions possibles d'un exemple simple : "$B\pm\sqrt{b}$".

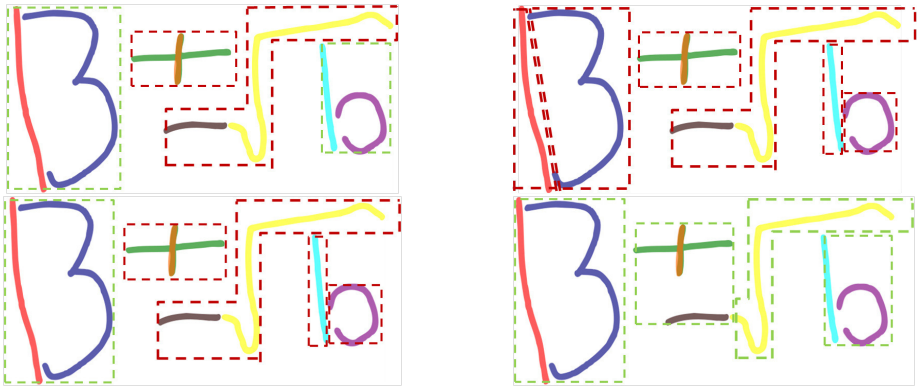
Face à la difficulté de cette tâche de segmentation et au regard des possibilités offertes par le signal, il est tentant d'imposer que l'élément de base ne soit plus un trait mais directement un symbole. En d'autres termes, on impose que chaque symbole soit tracé en un seul trait. C'est ce qui est fait dans divers travaux et dans divers domaines de la littérature qui admettent qu'un symbole est représenté par un seul trait [LM95, DHT00, PA03]. Dans ce cas, on ne s'intéresse plus à la formulation des hypothèses de symboles et on s'occupera de reconnaître directement les traits en leur associant les bonnes étiquettes. Pour être en mesure d'effectuer cet étiquetage de façon robuste, une contrainte forte est imposée lors de la pratique du geste donnant un trait (nécessité de reproduction de gestes très proches). Ceci réduit le caractère naturel de l'écriture et nécessite un apprentissage de l'utilisateur et non (en plus) du système. Cette approche a été mise en œuvre avec succès dans d'autres domaines que les EMs, comme par exemple la reconnaissance des partitions musicales manuscrites en-ligne [Cou96, MAGB05]. Mais, encore une fois, le nombre élevé de symboles, leur variabilité et leur similitude font que cette approche n'est pas adaptée aux EMs.

Les méthodes de segmentation les plus courantes dans le cadre de la reconnaissance des EMs reposent sur la construction de graphes ou de réseaux. Ce graphe contient toutes les hypothèses de segmentation qui satisfont un certain nombre de critères : la proximité géométrique, le nombre maximal de traits par groupement, mais aussi l'autorisation de sauts temporels. Le graphe ainsi construit contient les diverses partitions possibles (chacune contient tous les traits une seule fois) [LWL96, AMVG12].

1. http ://mathworld.wolfram.com/BellNumber.html



(a) Tracé original



(b) Quelques-unes parmi les 4140 partitions possibles

FIGURE 2.4 – Exemple d'une EM manuscrite en-ligne (2.4a) et quelques unes des partitions possibles (2.4b) : les segments valides sont entourés en vert, les mauvais quant à eux le sont en rouge

C'est ce type de techniques qui ont initié l'optimisation globale des systèmes de reconnaissance des EMs dont on parlera en section 2.6.

Une fois les hypothèses de segmentation formulées, celles-ci doivent être étiquetées, c'est l'étape de reconnaissance, qui est le propos de la section ci-après.

2.4 Reconnaissance des hypothèses de segmentation

Les objets délimités lors de la segmentation doivent maintenant être identifiés. Cette identité est donné par un lexique, plus connu sous la dénomination de **dictionnaire de labels**. Le système de classification a pour mission d'assigner l'étiquette la plus vraisemblable (et idéalement en associant un score ou une probabilité mesurant la confiance du système en sa réponse), voire fournir une liste d'étiquettes potentielles ordonnées de façon décroissante selon leur vraisemblance. Il s'agit là typiquement d'un problème de reconnaissance de formes. Toutefois, à la lumière de ce que nous avons présenté dans le chapitre 1, la mission du système de classification est ardue. Il doit combiner avec le nombre élevé de classes, la confusion inter-classes sans oublier la dispersion intra-classe. Qui plus est, il doit être capable de s'accommoder de situations ne correspondant à aucune classe lorsque des hypothèses invalides lui sont soumises.

Globalement, Chan et al. [CY00] ont identifié trois approches de reconnaissance de symboles mathématiques. Cela peut être fait par des approches basées sur :

1) Une association de motifs. Cela consiste à mettre en correspondance l'exemple à reconnaître avec tous les exemples d'une base de modèles préalablement étiquetés. Le label à associer à la fin est le plus proche respectivement à une mesure de similarité ou une distance. Les mesures les plus utilisées sont la **distance de Hausdorff** ou la **déformation temporelle dynamique** (dite **distance DTW** pour **D**ynamic **T**ime **W**arping en anglais). Des exemples de travaux dans le cas des EMs en-ligne peuvent être consultés dans [SW06] ou encore dans [RK09].

2) Une description structurelle. Dans ce cas, il est assumé qu'un symbole est défini par des primitives structurelles de base, connues sous le nom d'**allographes** [2] [TR07]. Ces primitives peuvent être des boucles, des courbes ascendantes/descendantes, des points, ... Par exemple 8 est composé de deux boucles, une en haut, l'autre en bas. Un symbole est de ce fait codé par une séquence de primitives et la classification se fait par mesure d'une distance entre séquences [YP93, EGS98]. Cette mesure est basée soit sur des méthodes syntaxiques à base de grammaires [BK88, Ram89], soit sur la comparaison de graphes (construits à partir des primitives extraites des symboles de la base de référence et de celui à reconnaître) [Bai88, Leb91], ou bien sur la comparaison de chaînes (également construite à partir des primitives des symboles de références et de celui à reconnaître)[BH84, AY92].

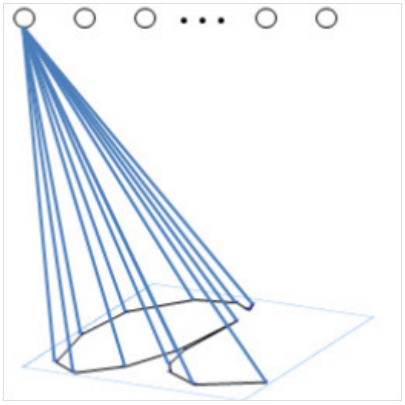
3) Des approches statistiques et des méthodes d'apprentissage. Il s'agit là des méthodes les plus utilisées pour les tâches de classification. C'est notamment grâce à leur pouvoir discriminant pour certaines ou génératif pour d'autres, que ces méthodes se sont généralisées. Leur usage est devenu très apprécié car elles permettent de gagner en précision et de réduire les temps de calcul comparativement aux méthodes basées sur la mise en correspondance de motifs. En effet, ces méthodes ont pour vocation de résumer les exemples de la base de référence (qui devient donc dans ce cas une base d'apprentissage) par un modèle. Ce modèle appliqué à une nouvelle donnée à étiqueter, fournit l'étiquette la plus probable sans pour autant avoir recours aux données de référence. Ces approches nécessitent souvent une phase de caractérisation de la forme à reconnaître. C'est pourquoi, on définit un jeu de caractéristiques permettant de discriminer suffisamment les formes entre elles. Nous reparlerons de cette étape à l'occasion de la présentation des systèmes utilisés au cours de nos travaux dans les chapitres 6 et 7.

Parmi les méthodes les plus employées dans la littérature on retrouve :

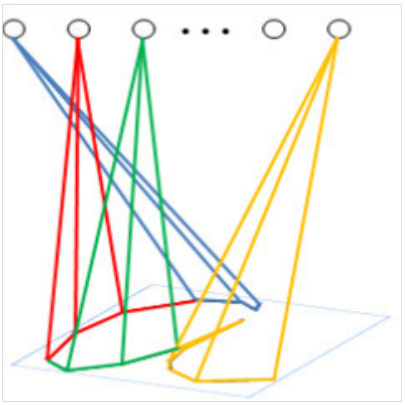
- **Les réseaux de neurones artificiels (*RNA*) :** Ce type de classifieurs est inspiré (une imitation ou une approximation) du fonctionnement du réseau neuronal biologique [MP43]. C'est la reproduction de la façon dont le réseau

2. Un allographe dans ce cas est à prendre dans le sens du procédé qui consiste à écrire différemment des mots (les symboles mathématiques dans notre cas). http ://-monsu.desiderio.free.fr/curiosites/allographe.html

neuronal humain perçoit et traite l'information. Un *RNA* est donc une interconnexion de neurones de base. Il est organisé en couches de trois types différents : la couche d'entrée qui est en charge de la capture de l'information extérieure, une couche de sortie en charge de restituer le résultat de la reconnaissance, et entre les deux un certain nombre de couches dites cachées [DHS01, CM02, Bis06].
Pour la reconnaissance de caractères manuscrits en-ligne, on retrouve plusieurs variantes de *RNA*. Les plus utilisés sont les **P**erceptrons **M**ulti-**C**ouches **PMC** (en anglais : **M**ulti **L**ayers **P**erceptron (**MLP**)) et les réseaux de neurones à convolution (dont la notation anglaise est **T**ime **D**elayed **N**eural **N**etwork (**TDNN**)) [Poi05]. Ces derniers sont une variante des *PMC* dont certaines couches sont dédiées à l'extraction de caractéristiques locales afin d'offrir à la partie PMC en charge de la classification une vision localisée du tracé (*cf.* l'illustration de la figure 2.5). Selon [Poi05], c'est cette extraction de l'information locale qui fait que ce type d'architecture soit très adapté au caractère temporel du signal manuscrit en-ligne.



(a) Ce que voit le PMC : vision globale, chaque neurone de la couche d'entrée voit toutes les caractéristiques de tous les points du symbole.
De même, chacun de ceux des couches supérieures voit tous les neurones des couches précédentes.



(b) Ce que voit le TDNN : vision locale, chaque neurone de la couche d'entrée ne voit que les caractéristiques des points du symbole capturés par une fenêtre glissante. De même, chacun de ceux des couches supérieures ne voit que les neurones des couches précédentes à l'intérieur d'une fenêtre glissante.

FIGURE 2.5 – Connexions du PMC *VS* connexions du TDNN [Poi05, Awa10]

Des travaux qui ont utilisé ce type de classifieurs dans le cadre de reconnaissance des symboles mathématiques sont par exemple [GAC+91, LY96], ou encore [AMVG12]. C'est sur ce type de classifieurs que nous avons basé le moteur de reconnaissance des symboles manuscrits en-ligne que nous avons utilisé et dont les résultats sont présentés dans ce manuscrit. Nous en donnerons une description plus détaillée à l'occasion du chapitre 6.

- **Les systèmes à vastes marges :** Il sont plus connus sous l'abréviation **SVM** de la notation anglaise **S**upport **V**ector **M**achines. Face au succès de ce type de classifieurs dans divers domaines, les SVM ont été explorés dans certains travaux de la reconnaissance des symboles mathématiques manuscrits en ligne [BHB02, KW07, KW08].
  La force des classifieurs de type *SVM* vient de leur principe. À savoir, construire un modèle dans lequel on puisse séparer le plus possible les différentes classes (d'où la dénomination vaste marge). Cette marge évaluée entre la frontière de séparation des classes et les échantillons les plus proches (dits **vecteurs de support**), une fois maximisée permet de définir le plan séparateur optimal [CV95, DHS01, Bis06]. Plus le nombre de classes est grand plus le modèle obtenu est complexe. Ajouté à la variabilité intra-classe et les confusions inter-classes, cela se traduit par des modèles encore plus complexes et lourds à apprendre. Une fois appris, ces systèmes, même optimaux, s'avèrent être gourmand en espace mémoire et temps de calcul. Ceci est la raison majeure limitant leur usage dans le cas des EMs, où bien souvent l'espace mémoire et le temps de calcul sont des critères importants. Ceci est d'autant vrai dans le cas des applications embarquées dont la demande actuelle est grandissante.
- **Les modèles statistiques :** Les **M**odèles de **M**arkov **C**achés (**MMC**), désignés par **HMM** en anglais pour Hidden Markov Models, sont des automates stochastiques qui permettent de modéliser les données séquentielles [Rab89]. L'écriture manuscrite (en-ligne, en particulier), étant un signal de ce type, se prête bien à des traitements à base de ces modèles. Le mode opératoire des *MMC* laisse penser que ce sont des outils très intéressants pour combiner à la fois la segmentation et la reconnaissance des symboles [TR07]. Énormément de travaux que nous avons retrouvés dans la littérature vont dans cette direction [BBNN93, SGH95, SK97, ÁSB12]. Nous reviendrons un peu plus en détail sur cette technique de classification à l'occasion du chapitre 7, dans la mesure où elle est le cœur du système de reconnaissance du signal de parole (un autre signal séquentiel) dans notre système.

Les symboles constituant l'EM étant identifiés (strokes inclus, informations géométriques, label(s) et score(s)associés(s)), il est maintenant nécessaire de découvrir la façon de les mettre en relation. Ceci est l'objet de la section qui va suivre.

2.5 Identification des relations spatiales et interprétation de l'EM

Par interprétation, on entend retrouver l'équivalent LaTeX ou *MathML* du tracé manuscrit initial. Le but est de considérer les symboles identifiés par les procédures expliquées dans les sections 2.3 et 2.4 et d'opérer successivement une phase d'analyse structurelle suivie d'une validation syntaxique [Tap04, Awa10]. Après ces deux

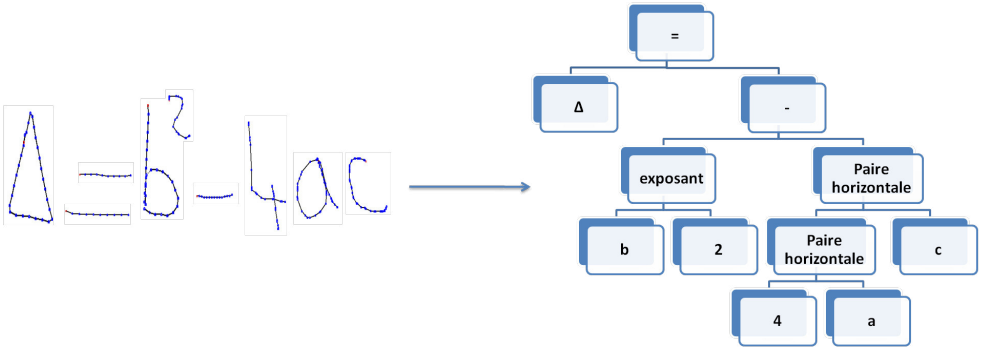


FIGURE 2.6 – Exemple d'arbre syntaxique donnant l'expression $\Delta = b^2 - 4ac$

traitements, un arbre syntaxique représentant l'expression manuscrite est obtenu. Ce dernier est convertible de façon triviale en une chaine LaTeX ou en un arbre *MathML* (voir figure 2.6).

Dans la majorité des approches rencontrées dans la littérature, l'arbre issu de cette analyse structuro-syntactique a pour nœuds les symboles et les informations géométriques qui les caractérisent. Les relations qui les lient les uns aux autres sont soit également représentées par d'autre nœuds ou soit portées par les arcs assurant la jonction entre les nœuds de l'arbre. Dans la suite nous citons quelques travaux majeurs ayant contribué à la mise en place de méthodes d'interprétation des EMs. Globalement trois principales approches se distinguent : les méthodes grammaticales probabilistes, les méthodes grammaticales graphiques et les méthodes grammaticales de coordonnées.

Méthodes basées sur la grammaire de coordonnées

Les systèmes de reconnaissance des EMs sont pilotés par des grammaires 2D. Celles-ci assurent la validité syntaxique de l'arbre résultant. C'est Anderson [And67] qui est considéré comme un des premiers à avoir abordé cette problématique. Il propose une grammaire dite de coordonnées. En effet, il s'appuie sur des règles (appliquées de façon descendante) dont l'élément central est l'opérateur, et définit autour de cet opérateur des zones où sont attendues les opérandes. Cette méthode bien que simple et précise, souffre du temps de calcul assez élevé requis, spécialement si énormément d'opérations implicites existent. Belaïd et al. [BH84], ont également utilisé une variante de cette grammaire en proposant une amélioration en opérant une analyse à la fois descendante et ascendante. Ceci permet, dans le cas d'une incertitude lors de la reconnaissance d'un symbole, de proposer une liste de meilleurs candidats potentiels avec leurs scores respectifs. L'information contextuelle, au cours de l'application des règles, contribue à la sélection du meilleur candidat de la liste précédente. C'est ce mode opératoire qui donne au système le pouvoir de faire le choix entre des symboles que seul le contexte permet de distinguer, telle que la parenthèse ouvrante "(" et un "C" majuscule.

Méthodes grammaticales probabilistes

On reste toujours dans la catégorie des méthodes basées sur des grammaires mais s'appuyant cette fois-ci sur les probabilités (ou coûts) des règles de production. On retrouve énormément de travaux à ce sujet dans l'état de l'art. Ces types de grammaires ont la réputation d'être très performantes. Toutefois, le fait que les probabilités des règles soient liées à la géométrie des éléments mis en cause dans la relation, rend cette méthode très sensible à la taille relative des éléments entre eux ainsi qu'au positionnement des lignes de base. Des exemples de travaux reportant l'usage de cette approche sont par exemple ceux de Chou [Cho89], ceux de Miller et Viola [MV98] ou plus récemment ceux de Awal et al. [AMVG12] et de Alvaro et al. [ÁSB12].

Méthodes grammaticales graphiques (ou plus exactement à base de graphe)

Les grammaires à base de graphe consiste à réécrire itérativement un graphe de départ grâce aux règles de production. Ces mêmes règles informent sur les identités des nœuds réécrits. le graphe initial étant composé des symboles reconnus comme faisant partie de l'EM. Parmi les travaux rencontrés opérant ainsi on citera Grabvec et Bolstein [GB95]. L'inconvénient majeur de cette approche est le temps de calcul considérable ainsi que l'influence de l'ordre d'application des règles impactant directement sur l'arbre final (obtention d'arbres différents).

2.6 Interaction reconnaissance-segmentation- interprétation

L'exécution des étapes de reconnaissance automatique du signal manuscrit (segmentation, reconnaissance et interprétation) est très sensible au problème de la propagation des erreurs. En effet, dans un enchainement séquentiel, une erreur de segmentation conduira forcément à une mauvaise reconnaissance et par conséquent à une mauvaise interprétation [Awa10]. C'est pour cela que de plus en plus de méthodes proposées visent à optimiser ces différentes procédures de façon conjointe autant que possible. Ceci va permettre de remettre en cause les propositions de l'une à partir de l'analyse faite dans une autre.

Les premiers travaux effectués dans cette direction, portaient sur l'interaction des étapes de segmentation et de reconnaissance. C'est le cas de Kosmala et al. [KRLP99] qui utilisent les Modèles de Markov Cachés (MMC) pour proposer directement les symboles qui offrent un meilleur compromis segmentation-reconnaissance. Dans [RSF+99], une approche basée sur de la programmation dynamique guidée par des coûts issus à la fois du module de reconnaissance et des informations structurelles liées aux groupes de strokes est proposée : la segmentation finale va être dans ce cas celle de coût minimal. Dans tous les cas, les symboles découverts sont directement exploités par le module en charge de l'interprétation.

Par la suite, une vision plus globale du problème de reconnaissance des EMs est apparue. Elle consiste à considérer une optimisation simultanée des trois étapes [RSTS06, AMVG09, AMVG12]. Cette orientation des méthodes se veut intuitive et perceptuelle. En effet, de façon naturelle pour un humain, interpréter un tracé d'une EM de façon visuelle active un mécanisme global ou les différentes étapes sont accomplies quasi-simultanément. Dans [RSTS06], c'est une grammaire probabiliste du type hors-contexte qui est exploitée pour une interaction complète des trois modules (segmentation, reconnaissance et interprétation). Les nœuds terminaux de la grammaire sont les hypothèses de segmentation auxquelles le classifieur a assigné un score. Les autres nœuds, portant l'information structurelle, se voient allouer des coûts du type géométrique. Ces coûts sont calculés en considérant les propriétés géométriques des symboles impliqués dans la relation. Cette approche suppose d'avoir à disposition un classifieur entraîné sur une base d'exemples de symboles isolés suffisamment représentatifs. Awal et al. [AMVG09, AMVG12], fondent leur système sur une approche similaire mais proposent un autre type d'interaction. En effet, ils proposent une méthode d'**apprentissage global** permettant d'entraîner le classifieur au sein du contexte réel, à savoir à partir d'EMs réelles, en rajoutant également une classe de rejet, qui prodigue au classifieur un pouvoir de rejeter des segmentations aberrantes.

2.7 Évaluation des systèmes de reconnaissance des EMs manuscrites en-ligne

Comme nous allons le voir dans la seconde partie de ce manuscrit (contributions de la thèse), l'évaluation de notre système bi-modal est similaire au cas général des systèmes de reconnaissance du tracé manuscrit en-ligne. De ce fait, nous présentons ici les diverses approches proposées à cet effet et en particulier celle dont nous allons faire usage dans ce manuscrit. L'évaluation dans le cas du tracé manuscrit en-ligne peut être accomplie à différentes échelles : globale, liée aux performances générales face au problème traité, ou locale, qui va être effectuée au niveau des sous-problèmes composant la tâche principale. Dans la suite, nous résumons les métriques les plus courantes dans la littérature et qui nous serviront dans les diverses évaluations rapportées dans ce manuscrit.

2.7.1 Évaluation globale : taux de reconnaissance au niveau EM

Dans le cas des EMs, l'évaluation globale va donner une idée de l'efficacité au niveau des expressions complètes. Il s'agit du taux de reconnaissance au niveau expression complète. Ce taux est calculé sur une base d'évaluation en comptant le nombre d'EMs bien reconnues ($NbEMvalides$) sur le nombre d'EMs total de la base d'évaluation ($NbEMtotal$). Dans cette formulation, il est bien évidemment indispensable d'être capable d'aligner de façon unique l'expression reconnue sur l'EM vérité terrain. C'est à cet effet qu'il est très important de disposer d'une représen-

tation canonique des EMs de référence et résultante à comparer. Ainsi, à l'occasion de la compétition *CROHME*, c'est sous une représentation canonique du format MathML que les EMs sont comparées et les participants ont pour consigne de respecter ce format. Ce taux de reconnaissance $TauxEM$ est donné par l'équation 2.2 :

$$TauxEM = \frac{NbEMvalides}{NbEMtotal} \tag{2.2}$$

Cette valeur de $TauxEM$ renseigne de façon assez peu fine, voire pénalisante sur les performances du système. Ceci est dû au fait qu'une EM est comptée dans la catégorie non valide même si un seul de ses symboles ou une seule de ces relations est mal reconnu(e), faisant abstraction de sa longueur ou de sa complexité. En d'autres termes, dans ces conditions, que l'EM contienne 50 symboles ou seulement 2 symboles, cette mesure sera la même si un seul de ces symboles est non retrouvé (alors que dans le premier cas, 98% des symboles sont reconnus, et la moitié dans l'autre). De même, deux évaluations sur deux bases présentant des EMs avec des complexités/longueurs différentes seront difficilement comparables.

Même avec cette lacune, énormément de travaux s'appuient sur ce critère pour l'évaluation de leurs systèmes [RSF+99, TSMS01, OIT01, RSTS06, SLS07, AMVG12, ÁSB12]. Cette métrique est d'ailleurs adoptée lors du classement des divers systèmes participant aux différentes éditions de la compétition *CROHME*. En effet, même si d'autres mesures ont été effectuées (plus locales qu'on présentera dans la suite), la désignation du vainqueur est à chaque fois basée sur celui qui renvoie le meilleur taux de reconnaissance au niveau expression complète [MVGG+11, MVGK+12, MVGZ+13].

2.7.2 Évaluation locale

Dans ce type de métriques, les systèmes sont évalués non pas sur le succès ou non de la tâche qui leur est assignée, en sa totalité, mais sur leurs performances au niveau des étapes intermédiaires conduisant au résultat final. Dans le cas de l'écrit par exemple, les étapes segmentation-reconnaissance-interprétation (*cf.* sections précédentes de ce chapitre) peuvent être évaluées de façon locale. En effet, une erreur de segmentation peut de façon inévitable conduire à l'échec de la reconnaissance de l'EM ($TauxEM = 0$). Toutefois, il serait intéressant de savoir à quel point le système s'est éloigné de la bonne solution. Pour cela savoir combien de bonnes hypothèses de symboles sont formulées au sein de la solution est une bonne mesure. Cette mesure dite **taux de segmentation**, notée $TauxSeg$, est utilisée dans beaucoup de travaux également [MV98, RSF+99, RSTS06, GHLJ07, AMVG12]. Elle est donnée par le rapport du nombre de bon groupements de traits $NbBonneSeg$ (hypothèses de symboles) et du nombre de groupements total présents dans l'EM/ou la base d'évaluation (mesure homogène au taux de rappel du système) $NbTotalSeg$ (voir équation 2.3).

$$TauxSeg = \frac{NbBonneSeg}{NbTotalSeg} \tag{2.3}$$

Le taux de segmentation renseigne donc sur la capacité du système à accomplir au mieux la tâche de segmentation.

Une autre métrique très utilisée, de par sa simplicité de mise en œuvre, a pour vocation d'évaluer le classifieur en charge d'étiqueter les hypothèses de symboles formulées. Elle donne le ratio des symboles bien reconnus $NbBonneRecoSymb$ comparativement au nombre total de symboles $NbTotalSymb$. On notera ce nouveau critère d'évaluation $TauxRecoSymb$, il est donné par l'équation 2.4.

$$TauxRecoSymb = \frac{NbBonneRecoSymb}{NbTotalSymb} \tag{2.4}$$

Deux variantes de cette mesure de taux de reconnaissance au niveau symbole peuvent être distinguées, et cela en fonction de la façon de considérer $NbTotalSymb$. La première est de prendre $NbTotalSymb$ comme étant l'ensemble des symboles que contient l'EM (ou la base). Dans ce cas, $TauxRecoSymb$ renseigne non seulement sur le pouvoir de classification du classifieur mais aussi sur le degré du succès de la tâche de segmentation [MV98, RSF+99, RSTS06, GHLJ07, AMVG12]. La seconde variante est de considérer que $NbTotalSymb$ donne uniquement le nombre de bonnes hypothèses de segmentation. Dans ce deuxième cas, on ne mesure effectivement que la qualité du classifieur [Awa10, MVGG+11, MVGK+12, MVGZ+13].

D'autres mesures ont été définies pour rendre compte de la robustesse du module en charge de l'interprétation des EMs, même si elles sont rares. Il s'agit, pour la plupart des travaux recensés [RSF+99, KFDY01, KRLK09, Awa10, MVGG+11, MVGK+12, MVGZ+13], de compter le nombre de relations bien retrouvées (*c.a.d*, interpréter correctement les relations entre les symboles sans considérer les étiquettes des symboles) $NbBonneRecoRel$ parmi toutes les relations existantes $NbTotalRel$. Cette mesure, notée $TauxRecoRel$, est donnée par l'équation 2.5 :

$$TauxRecoRel = \frac{NbBonneRecoRel}{NbTotalRel} \tag{2.5}$$

Awal et al. [Awa10, AMVG12] ont proposé une extension de la métrique de l'équation 2.5, en distinguant deux variantes. La première considère effectivement les $NbBonneRecoRel$ comme le nombre de **relations correctes**. Cela veut dire qu'en plus du type de la relation, les symboles qui y sont impliqués sont les mêmes dans les deux arbres, résultat et vérité terrain. La seconde variante, considère uniquement les étiquettes des relations et fait abstraction des symboles impliqués dans la relation ($NbBonneRecoRel$ est dans ce cas le nombre de toutes les relations retrouvées). Cette deuxième formulation rend compte effectivement de la qualité de l'interprétation, l'identité des symboles étant en grande partie assurée par le classifieur. Dans ce cas on parle de **relations retrouvées**. Ces deux types de mesure ont été utilisés dans le cadre des compagnes de la compétition *CROHME* [MVGG+11, MVGK+12, MVGZ+13].

Comparativement au cas des métriques sur les symboles et sur la totalité de l'EM, les métriques sur l'évaluation des relations spatiales sont moins présentes dans littérature. Cela est principalement dû au fait que pour arriver à ce genre de mesure, il est indispensable d'aligner les arbres résultat et vérité terrain représentant l'EM tel que nous l'avons mentionné en début de cette section. Ce problème, d'alignement d'arbres, est connu pour être *NP-difficile* [Hul96].

Plus récemment, Zanibbi et al [ZPM+11, ZMVG13] ont proposé plusieurs métriques d'évaluation basées sur les éléments de base composant l'EM (traits élémentaires dans le cas du signal en-ligne). En effet les EMs à comparer (vérité terrain et résultat) sont dans un premier temps représentées sous forme de graphes d'étiquettes (*Label graphs* en anglais). Ce dernier représente une structure au niveau trait. Chaque trait est représenté par un nœud étiqueté par le même label que le symbole auquel il appartient. Chaque couple de nœuds (*ie* couple de traits) est lié par une paire d'arcs orientés. L'étiquette associée à chaque arc indique soit que les deux traits font partie du même symbole (pas de relation spatiale entre eux) ; soit qu'ils sont liés à travers une relation spatiale (à droite, à gauche, en bas, en haut, dans, indice, exposant) et à ce moment là, le label de l'arc est le type de la relation ; soit est sans étiquette.

En se basant sur cette structure, la mise en correspondance du graphe de la vérité terrain et du graphe du résultat est directe car guidée par l'unicité physique des traits (on sait quel nœud est à comparer avec quel nœud). La distance de *Hamming* [Ham50] est ensuite utilisée pour rendre compte des différences pouvant exister. Les métriques définies et utilisées lors de la troisième édition de la compétition *CROHME* [MVGZ+13] dont les scripts sont librement disponibles en ligne[3], sont comme suit :

- ΔC : nombre de labels de traits différents
- $\Delta L = \Delta S + \Delta R$: nombre de labels d'arcs différents. Il est donné par le cumul du nombre de mauvaises segmentations (combinaisons de traits qui ne font pas partie du même symbole ou oubli de certains traits du même symbole, sous/sur-segmentation) ΔS et du nombre de mauvaises étiquettes de relation ΔR
- $\Delta B = \Delta C + \Delta L$: est la distance globale entre les deux structures de l'EM.

À partir de ces métriques sont définies des versions normalisées, comme suit (pour n traits élémentaires de l'EM) :

- le pourcentage de labels corrects du graphe entier.

$$\Delta B_n = \frac{\Delta B}{n^2} \tag{2.6}$$

- l'erreur moyenne sur l'ensemble des tâches segmentation, classification et interprétation.

$$\Delta E = \frac{\frac{\Delta C}{n} + \sqrt{\frac{\Delta S}{n(n-1)}} + \sqrt{\frac{\Delta L}{n(n-1)}}}{3} \Delta S + \Delta R \tag{2.7}$$

3. http ://www.cs.rit.edu/∼prl/Software.html

2.8 Bilan

Ce chapitre a mis en lumière les solutions apportées par la communauté scientifique pour la reconnaissance des expressions mathématiques manuscrites, ouvrant ainsi la voie pour des systèmes automatiques facilitant les tâches que certains sont amenés à exécuter de façon répétitive : saisie des formules mathématiques, recherche sur internet de documents contenant des EMs spécifiques, enseignement interactif des mathématiques, ... La barrière principale à laquelle ces systèmes font face est liée aux spécificités évoquées dans le chapitre 1, section 1.2. Malheureusement certaines de ces ambiguïtés ne peuvent être levées car elles sont intrinsèques au signal manuscrit lui-même. Dans ces cas, on ne peut que constater que l'on a atteint les limites d'une telle modalité. Ceci suggère le recours à des sources d'information externes pouvant aider le processus de reconnaissance. Le signal audio, issu de la parole, semble le plus approprié. Nous allons procéder dans le prochain chapitre à la présentation de cette modalité et voir les apports qu'elle peut avoir.

Chapitre 3

Reconnaissance des expressions mathématiques parlées

Sommaire

Au cours de ce chapitre c'est la problématique d'interprétation des expressions mathématiques dictées qui est exposée. Après la présentation du principe général des systèmes de reconnaissance de la parole, l'accent est mis sur les ajouts nécessaires à ces derniers pour être en mesure d'interpréter le texte en expression mathématique.

3.1 Introduction

La philosophie d'un système de transcription automatique de la parole (STAP) la plus utilisée de nos jours est celle définie par Jelinek et al. dans les années 70 [Jel76]. Il s'agit d'une approche statistique tirée de la discipline de la théorie d'information où dans ce cas, on cherche à découvrir une séquence de mots à partir de séquences d'observations acoustiques. Le but est similaire à celui d'un système de reconnaissance de l'écriture manuscrite. La différence principale est qu'au lieu de découvrir la séquence de mots à partir de tracés manuscrits élémentaires, c'est à partir des séquences acoustiques que les mots sont retrouvés. Cette différence vient bien évidemment de la nature du signal traité. Cette discipline, au même titre que celle de la reconnaissance de l'écriture, rencontre un engouement prononcé de la part de la communauté scientifique, mais aussi du grand public [HCF+06]. Cet intérêt, tout comme dans le cas de l'écrit, est dû au caractère intuitif et pratique de la modalité de la parole. Il ne cesse de s'accroître du fait des possibilités offertes de nos jours. Le *STAP* permet par exemple de converser avec sa machine en cas de handicap, temporaire ou permanent, empêchant l'usage des mains et/ou des yeux. Il facilite également l'usage de certaines applications nécessitant une expertise de l'utilisateur, surtout si ce dernier est novice. Plus récemment, avec les développements technologiques, l'accès au web, la sécurisation des systèmes basés sur l'empreinte vocale mais aussi la navigation web sont des applications très en vogue [HCF+06].

Dans la section 3.2.1 nous allons décrire brièvement le principe de la transcription automatique de la parole et en donner le fonctionnement global. On dédiera la section 3.2.2 à l'interprétation de la transcription donnée par le *STAP* en langage mathématique.

3.2 Reconnaissance automatique des expressions mathématiques à partir de leur dictée

Richard Fateman entame son article *"How can we speak math ?"* [Fat98] en disant qu'il très probable que les gens transmettraient les expressions mathématiques à leurs machines plus efficacement (plus rapidement et avec une plus grande précision) en les dictant qu'au moyen de tablettes et de stylets. Cela paraît assez surprenant car la vision 2D d'une EM est intrinsèquement contenue dans le cas du signal manuscrit, alors qu'elle n'a d'existence dans le cas de la parole qu'après une analyse du texte issu du STAP. Toutefois, Richard Fateman argumente son propos en rapportant les diverses expérimentations menées au sujet de différentes configurations (modalités) d'interface d'entrée faisant intervenir l'écriture manuscrite notamment. Par contre, il est tout de suite rapporté que l'usage de la parole spontanée pour accomplir cette tâche est dans la plupart des cas infructueuse et nécessite soit l'intervention d'une modalité supplémentaire, soit l'imposition de règles de diction. Il continue en disant que non seulement, il faut ces contraintes pour être en mesure de déduire correctement la bonne EM associée au texte fourni par le

STAP, mais il précise que dicter une expression n'est pas toujours chose facile. En effet la complexité des EMs et/ou le manque d'entraînement du locuteur vont être des obstacles certains pour cette tâche de traduction du texte (1D) vers l'EM (2D). Ces deux contraintes n'en sont pas pour le cas de la modalité écrite par exemple. Aussi complexe que soit l'EM et quelque soit le niveau du scripteur, ce dernier saura toujours la recopier efficacement (plus ou moins difficilement). On reviendra sur ces aspects dans la section 3.2.2, mais avant commençons par présenter les systèmes de reconnaissance de la parole fournissant la description textuelle de l'EM (information 1D).

3.2.1 Reconnaissance automatique de la parole

L'évolution des systèmes de reconnaissance vocale est rapportée dans [BDMD+07, AK10], à commencer par les premières études théoriques sur le sujet dans les années cinquante [DBB52] jusqu'aux systèmes existants actuellement [CER+07]. Les différents *STAP* se distinguent les uns des autres principalement à deux niveaux. Le premier concerne les caractéristiques utilisées pour la reconnaissance. Le second niveau est lié aux techniques sur lesquelles repose le module de décodage de la séquence de mots ($HMM, DTW, ...$) [BDMD+07]. On reviendra sur ces deux aspects dans les sections qui vont suivre. Avant cela, on donne sur la figure 3.1 une représentation de l'architecture globale d'un système de reconnaissance automatique de la parole.

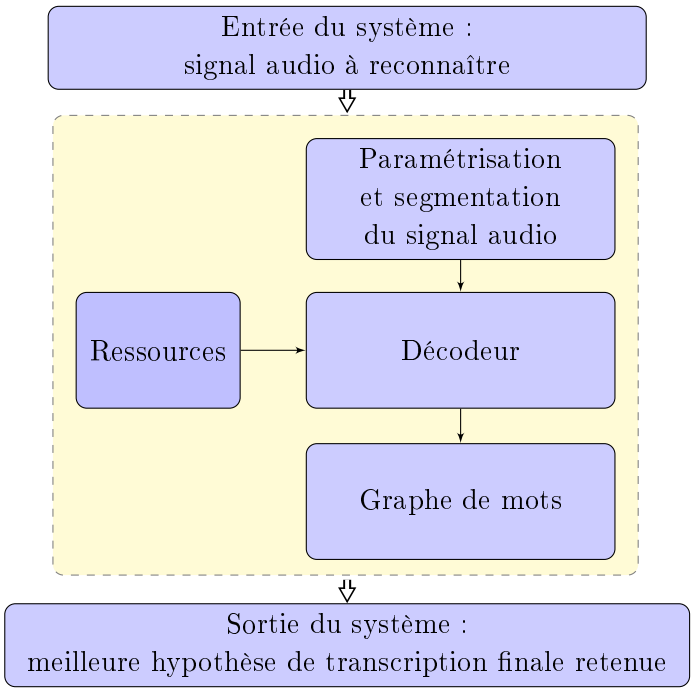


FIGURE 3.1 – Schéma de principe d'un système de reconnaissance automatique de la parole

A présent, nous allons décrire chacun des sous-blocs composant le système représenté en figure 3.1, nous donnerons à chaque fois des exemples du système de reconnaissance *Sphinx* [CER+07], car le système que nous avons utilisé dans les travaux rapportés ici s'en est largement inspiré.

3.2.1.1 Paramétrisation et segmentation du signal de parole [Bru95]

L'extraction de la séquence de mots servant à construire la transcription finale ne se fait pas sur le signal temporel brut de la parole. En effet cet espace de représentation (déformation de la pression de l'air engendrée par le locuteur dans le temps) est très peu discriminant. Le passage à des représentations assurant une représentation par des caractéristiques pertinentes du signal est requis. À coté de cela, il est notamment nécessaire de découper le signal audio par trame, c'est généralement fait sur des fenêtres de taille aux alentours de 25 ms assurant une quasi-stationnarité du signal. Un recouvrement des fenêtres successives d'environ 60% est recommandé pour s'assurer de disposer de toutes les séquences de mots possibles parmi lesquelles le tri sera fait pour sortir la solution finale [HCF+06]. Classiquement, pour passer du signal temporel aux vecteurs de caractéristiques on doit accomplir les tâches suivantes (également reprises sur le schéma de la figure 3.2) :

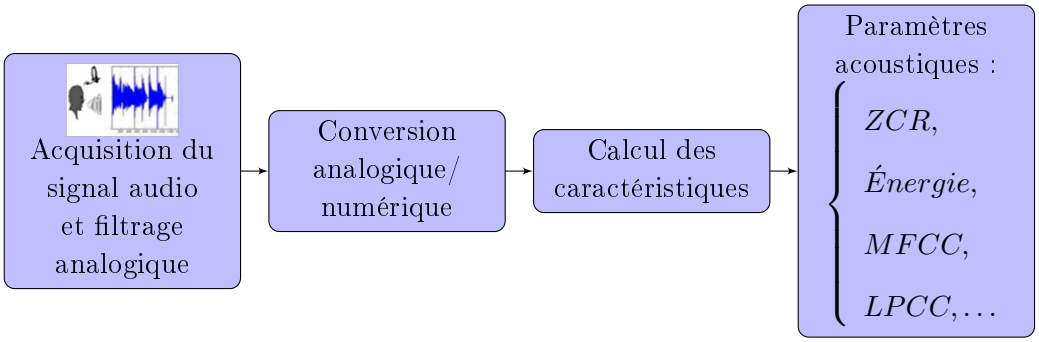


FIGURE 3.2 – Schéma décrivant le passage de la description temporelle du signal au domaine des caractéristiques

1) **Paramétrisation du signal.** La paramétrisation du signal, encore appelée pré-traitement acoustique, consiste en un filtrage analogique, suivi d'une conversion analogique/numérique (échantillonnage et quantification). Il s'agit de préparer le signal pour l'étape d'extraction des coefficients (caractéristiques) décrite ci-après.
2) **Les caractéristiques utilisées.** Les caractéristiques utilisées en reconnaissance de la parole sont similaires à celles utilisées en reconnaissance de symboles manuscrits en ligne. En effet, pour le signal écrit dans notre cas, les dérivées premières et secondes des positions sont considérées. En parole également, des caractéristiques généralement fréquentielles associées à leurs dérivées premières et secondes sont considérées. Ces caractéristiques fréquentielles peuvent être les coefficients Cepstraux (*MFCC* pour Mel-Filtered Cepstral

Coefficients) [DBB52], les coefficients de prédiction linéaire (*LPC* pour Linear Predictive Coding) [AH71], ou encore les coefficients de prédiction linéaire perceptuels (*PLP* pour Perceptual Linear Prediction) [Her90]. Les coefficients *MFCC* sont les plus utilisés. C'est précisément ces coefficients qui seront utilisés dans notre système, nous en donnons une brève description (plus de détails seront apportés dans le chapitre 6).
La chaîne de calcul de ces coefficients *MFCC*, au nombre de 39 par fenêtre du signal, suit un certain nombre d'étapes inspirées par le système auditif humain. Ce sont les suivantes :

- Pré-accentuation du signal : rehausser le signal au niveau des hautes fréquences (traitement sur tout le signal).
  Tous les traitements qui vont suivre sont appliqués sur des fenêtres d'environ $25ms$ [Bru95].
- Calculer la FFT (Transformée de Fourier rapide) sur 512 points.
- Convertir le spectre en échelle de *Mel* (échelle adaptée à la perception humaine). Cette nouvelle échelle, notée m est déduite de la fréquence f, qui est exprimée en Hertz (Hz) comme donné par l'équation 3.1 :

$$m = 2595 \times \log(1 + f/700) \tag{3.1}$$

- Prendre le logarithme de ce spectre en échelle de Mel
- Calculer la transformée cosinus (DCT) du log-spectre obtenu précédemment. Puis retenir les 13 premiers coefficients.
- Calculer les dérivées premières de ces 13 coefficients et leurs dérivées secondes à partir du signal issu de la DCT du log-spectre.

3) Segmentation. Les données sont interprétées en termes d'unités phonétiques ou de segments minimaux proches d'unités linguistiques. Celles-ci sont modélisées à l'aide de connaissances d'experts et constituent ainsi une base de faits [Bru95].

3.2.1.2 Décodeur [Duf10]

Il a pour tâche, en s'appuyant sur les ressources (modèles acoustiques, modèles de langage et dictionnaire phonétisé ou de prononciation), que nous allons présenter par la suite, de retrouver la séquence de mots (texte) correspondante. Cette tâche est accomplie à plusieurs niveaux, chacun d'eux faisant appel à une ressource en particulier (voir figure 3.3). En bref, les trames acoustiques sont dans un premier temps utilisées par l'algorithme de décodage (typiquement l'algorithme de Viterbi) pour déduire un graphe d'états (dit aussi graphe de phonèmes) en utilisant le modèle acoustique. Après cela, en s'appuyant sur le graphe d'états précédent, un graphe de mot est obtenu grâce au dictionnaire de prononciation. C'est à partir de ce graphe de mots et en faisant appel au modèle de langage qu'est obtenue la ou les meilleure(s) transcription(s).

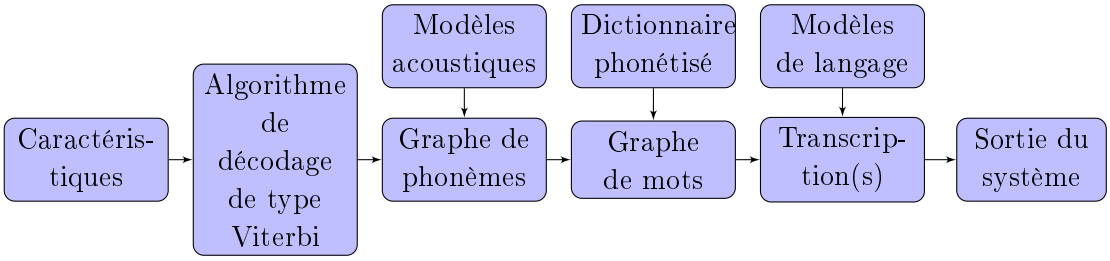


FIGURE 3.3 – Niveaux d'exploitation des ressources par le décodeur

3.2.1.3 Les ressources

1) Modèles acoustiques. Une fois le signal audio caractérisé, ce dernier est représenté par des trames élémentaires proches des unités linguistiques de base : les *phonèmes*. Autrement dit, en partant d'observations (les caractéristiques), on est capable de retrouver les états (phonèmes) qui ont permis ces observations.

Cette formulation est analogue à ce qui est fait dans le domaine de reconnaissance de formes à travers les Modèles de Markov Cachés (*MMC*). Dans ces modèles, les états cachés responsables des observations sont déduits grâce à une modélisation statistique du phénomène traité. Les *MMC* sont bien adaptés, de par leur définition, aux problèmes de reconnaissance à aspect temporel tel que la reconnaissance du geste, de l'écriture manuscrite mais également de la parole.

Dans le cas de Sphinx, les modèles acoustiques sont justement de type *MMC* [CER+07, Rab89]. Plus précisément ce sont des *MMC* à temps continu, ou chaque observation est caractérisée par une densité de probabilité. Cette dernière est modélisée par un mélange de 8 gaussiennes. Trois ou cinq états par *MMC* sont considérés. La topologie adoptée est celle de *Bakis*, dite également modèle gauche-droit autorisant un saut d'état (*cf.* figure 3.4). Ce type de topologie est appelé ainsi parce qu'elle n'autorise aucune transition d'un état vers un autre d'indice inférieur : les états qui se succèdent ont donc des indices égaux ou supérieurs aux précédents. Une fois dans le dernier état, le système est condamné à y rester. C'est pourquoi la probabilité initiale du premier état est fixée à 1, celles des autres étant égales à 0. Les états cachés, dans ce cas, sont les unités linguistiques de base : les phonèmes (au nombre de m) qu'on notera par $W = \{w_1...w_m\}$. Les vecteurs de caractéristiques quant à eux représentent les observations (au nombre de n et 1 vecteur = 39 coefficients *MCFF* par exemple), on les notera par $O = \{o_1...o_n\}$. Statistiquement parlant, on cherche à trouver la séquence d'états $\hat{W}$ qui maximise la probabilité à *posteriori* $P(W/O)$:

$$\hat{W} = argmax(P(W/O)), \tag{3.2}$$

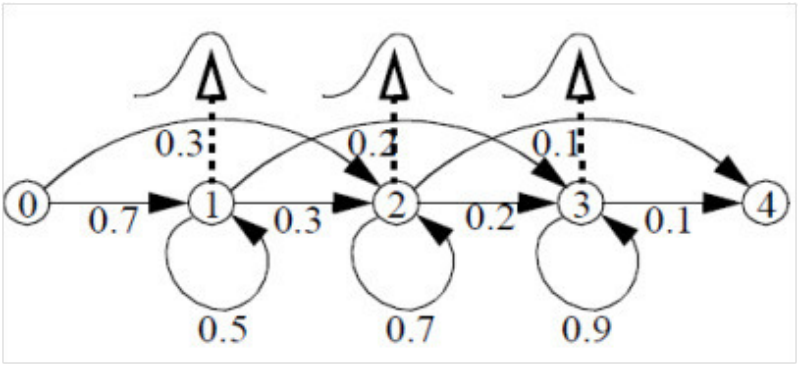


FIGURE 3.4 – Exemple d'un *MMC* à 5 états, avec un état initial et un état final qui n'émettent pas d'observations. Chaque cercle est un état. Les transitions permises sont représentées par des flèches, de sorte que si par exemple à l'instant t on est à l'état 1 alors, à l'instant $t+1$, il y a une probabilité de 0.5 de rester dans le même état, une probabilité de 0.3 de passer à l'état 2, et une probabilité de 0.2 de passer à l'état 3. Les flèches en pointillé montrent les densités de probabilité associées à chaque état, la probabilité des observations étant spécifiée par cette dernière [RSRS00, CER+07]

qui devient, en vertu de la règle de Bayes (équation 3.3) :

$$\hat{W} = argmax\left(\frac{P(W)P(O/W)}{P(O)}\right) \tag{3.3}$$

Comme la séquence d'observation acoustique (O) est fixée et est constante par rapport à W, l'équation 3.3 devient de ce fait :

$$\hat{W} = argmax(P(W)P(O/W)). \tag{3.4}$$

Le modèle probabiliste qui permet d'avoir $P(O/W)$ est donc le modèle acoustique. La valeur de $P(W)$ est donnée par le modèle de langage (voir plus loin). Dans [CER+07], un *MMC* est construit par phonème tout en tenant compte de son contexte gauche et de son contexte droit (*ie.* du phonème qui le précède et de celui qui le suit).
Ce triplet constitue le tri-phone. Ceci signifie qu'un phonème donné (par exemple le son *[a]*) aura plusieurs modèles dépendant de son contexte. Pour une meilleure finesse des modèles, la position du phonème au sein du tri-phone est aussi utilisée (début, milieu, fin ou même isolé).

2) Phonétisation. La phonétisation joue un rôle important dans la mesure où elle conditionne la qualité des modèles acoustiques. Elle consiste à caractériser chaque mot du vocabulaire par la séquence de phonèmes qui le constitue, c'est-à-dire sa *prononciation*. Un mot pouvant être prononcé de différentes manières, toutes ces dernières doivent être mentionnées dans le dictionnaire. Ce dictionnaire phonétisé va donc associer à chaque vocable, une (ou plusieurs)

chaîne(s) de phonèmes. Ceci étant fait, un alignement phonème/signal devient possible. Toutefois, il peut arriver que plusieurs phonétisations existent pour un mot donné. Ces phonétisations sont de longueur variables et ce, en fonction de la présence ou non de liaisons entre phonèmes. La question de la phonétisation à considérer se pose à ce moment précis. Si on considère les phonétisations courtes, les *MMC* des phonèmes proches d'une liaison seront bruités. Au contraire le choix de celles qui sont longues va dégrader les *MMC* associés aux phonèmes utilisés pour les liaisons lorsque celles-ci ne sont pas présentes. Laquelle faut-il choisir, à ce moment ?
Pour étaler l'erreur sur la plus grande partie possible des phonèmes constituant le mot en question, et équilibrer ainsi au mieux la dégradation prévisible des *MMC*, la phonétisation courte est en premier lieu utilisée [CER+07]. Puis, une fois les premiers modèles acoustiques disponibles, il est possible (les outils sont disponibles sous Sphinx) de raffiner ces derniers en exploitant le dictionnaire de prononciation pour chercher les meilleurs alignements phonème/signal et déduire ainsi les modèles acoustiques finaux.

3) Modèles de langage. Ce sont des modèles probabilistes. Ils sont utilisés, en guise de post-traitement, pour trancher entre les séquences de mots probables en attribuant un score. Ce score n'est rien d'autre que la probabilité qu'une séquence de mot apparaisse. Les *modèles n-grammes* sont les plus utilisés à cette fin. En pratique, n ne dépasse que rarement 4 (*Sphinx* tolère des modèles bi-grammes ou tri-grammes). Il peut arriver qu'un certain nombre de n-grammes n'apparaissent jamais dans les données d'apprentissage (quantité de données insuffisante). Pour pallier cela, les techniques de lissage permettent d'attribuer des probabilités non nulles à des n-grammes qui ne sont pas forcement dans le corpus d'apprentissage [CG96].

À ce stade nous avons présenté le comportement du système type de transcription automatique de la parole dédié en priorité au langage naturel (1D par essence). En effet la solution proposée en sortie est un texte qui traduit aussi fidèlement que possible la dictée enregistrée (selon la robustesse du système bien sûr, mais aussi la qualité du signal : présence ou absence de bruit, clarté ou rapidité de l'élocution du locuteur, qualité du matériel d'enregistrement, ...). L'exploitation de ce type de systèmes pour des langages $2D$ tel que les mathématiques est loin d'être triviale. Dans la suite nous abordons cette question.

3.2.2 Interprétation des expressions mathématiques à partir de la transcription automatique

À la lumière des propos de Richard Fateman, considéré comme l'un de ceux qui ont le plus travaillé sur le rôle de la parole dans le langage mathématique et un des premiers à avoir proposé des solutions à cette fin (*cf.* section 3.1), mais aussi au panorama, sur les mathématiques parlées, qu'il en fait dans [Fat98, Fat06], il est clair que la parole peut avoir un rôle important à jouer dans la reconnaissance (ou du moins dans l'aide à la reconnaissance) des langages graphiques, tels que les

mathématiques.

Cette modalité, a été initialement sollicitée pour la dictée de formules mathématiques par les machines à partir de leur chaine LaTeX ou de l'arbre *MathML* soit pour des personnes atteintes de cécité, comme c'est le cas pour l'application *AsTeR* [Ram98], ou pour des fins pédagogiques (apprentissage de la prononciation des EMs) comme *MathPlayer* [1] de *`design Science´* qui est aussi à l'origine de l'interface interactive d'édition via le clavier-souris, *MathType* (*ref.* figure 1.2a). Dans ce cas, c'est une autre problématique du traitement automatique de la parole qui est explorée : la synthèse de la parole à partir du texte (dite *TTS*, pour **T**ext **T**o **S**peech). Les dictées formulées par ces outils sont parfois peu représentatives de la sémantique de l'EM, en particulier pour le cas de *AsTeR* qui se base uniquement sur la chaine LaTeX de l'EM et ne prend pas en compte la sémantique [Fat98].

Le problème inverse, consistant à retrouver l'EM (chaine LaTeX , arbre *MathML*, ou autres...), n'est que très peu exploré. À côté des premiers travaux initiés par Richard Fateman [Fat98, GJSF04] (en proposant l'outil *Math Speak & Write*), qui exploite le *STAP* de *Microsoft*, en définissant une grammaire adaptée au cas des EMs, on retrouve principalement deux autres systèmes. Le premier, académique, faisant également appel au *SDK* de *TAP* de *Microsoft* [2] est baptisé **CamMath** [EB07]. Le second système, commercial, dénommé *MathTalk*, quant à lui, repose sur l'un des *STAP* commercialisés le plus performant, le *Dragon Naturally Speaking* [3] [WHP+09].

Le point commun de tous ces systèmes est de procéder en deux passes. La première concerne l'usage d'un *STAP* pour fournir le texte décrivant l'EM. La seconde, est la conversion de ce texte (1D) en une EM dans l'espace 2D.

Dans le premier module en charge de la transcription automatique de la parole, des outils très aboutis sont disponibles provenant directement de la reconnaissance automatique de la parole en général. Pour cette fin, on retrouve, à coté des systèmes commercialisés cités ci-haut (le *SDK* de *TAP* de *Microsoft* et le *Dragon Naturally Speaking*), des systèmes à base de logiciels libres, parmi lesquels : *Sphinx* (un des plus utilisés à travers le monde) [CER+07], *HTK* [4] [You06], *Julius* [5] ou encore *ISIP* [6].

Le second module destiné à retrouver l'EM, quant à lui, est en charge d'une tâche qui est loin d'être triviale. Ceci est notamment dû aux ambiguïtés évoquées à la section 1.2 du chapitre 1 et rapportées par l'exemple de la figure 1.10 en particulier. Dans [MS03], une solution pour passer du langage naturel (le texte 1D en anglais) au langage mathématique est implémentée. Cette solution proposée gère les ambiguïtés spatiales notamment en utilisant une syntaxe parenthésée. C'est à dire que si des éléments de l'EM peuvent avoir plusieurs interprétations possibles au niveau relation, des parenthèses sont utilisées pour en privilégier certaines. Par exemple pour réduire l'ambiguïté de la phrase "*a plus b sur c plus d*" de l'exemple

1. http ://www.dessci.com/en/products/mathplayer/
2. http ://www.microsoft.com/en-us/download/details.aspx?id=14373
3. http ://www.nuance.fr/for-individuals/by-product/dragon-for-pc/index.htm
4. http ://htk.eng.cam.ac.uk/
5. http ://julius.sourceforge.jp/en/
6. http ://www.isip.piconepress.com/projects/speech/software/

de la figure 1.10, celle-ci est réécrite de la façon suivante : "*(a plus b) sur (c plus d)*". De la sorte, l'interprétation de l'EM est unique. Cette solution est efficace quand la description est faite par l'humain est non fournie par le *STAP*. Ce dernier ne peut formuler ce genre de description que si le locuteur le fait durant sa dictée. Ceci revient à limiter les possibilités de la modalité audio et rend la diction moins naturelle et pesante sur l'utilisateur d'un tel système (possibilités d'erreurs dues à l'oubli ou la prévalence du caractère naturel durant la dictée). Le système *Math Speak & Write*, ne se confronte pas à ce genre de problèmes, du fait qu'il ne gère que les expressions linéaires (proche du texte) autorisant des exposants ou indices de type symboles et exploitant des pointeurs par le biais d'un signal manuscrit ou par la souris par exemple pour aider à corriger des éventuelles erreurs. *MathTalk*, quant à lui, impose l'usage de pauses temporelles variables inter-symboles mais surtout inter-éléments définissant une relation. Par exemple pour l'équation $\frac{a+b}{c+d}$, le temps de pause entre le numérateur et le dénominateur va être plus important que celui entre les différents symboles. *CamMath*, de son côté, fait appel, en plus des pauses (toutefois moins longues et moins fréquentes que dans le cas de *MathTalk*), à des règles de dictée pour spécifier par exemple la passage du numérateur au dénominateur puis pour exprimer la fin du dénominateur. L'exemple précédent s'exprimerait de la façon suivante : "*fraction <pause> a plus b <pause> bas <pause> c plus d <pause> fin fraction*".

3.3 Évaluation des systèmes de reconnaissance de la parole

La tâche du système de transcription automatique de la parole dans notre architecture globale (*cf.* chapitre 7) est uniquement de fournir la description textuelle de l'EM donnée par le signal audio. Nous mesurerons de ce fait la fiabilité de ce système en calculant une métrique qui est d'un usage très répandu dans le domaine de la parole [MMD+05]. Il s'agit du taux d'erreur mot (*WER*) duquel on peut immédiatement déduire le taux de reconnaissance en mot (*cf.* équation 3.5), qu'on notera $TauxMots$. C'est grâce à la distance d'édition (dite aussi distance de *Levenshtein* [Lev66]) qu'est calculé le *WER*. Ce dernier est proportionnel au rapport du nombre de mots mal reconnus $NbMotsMalReconnus$ et du nombre total de mots dictés réellement (vérité terrain) $NbMotsTotal$. On considère trois types d'erreur de reconnaissance possibles, tous trois affectés d'un même coût. Il s'agit de : S le nombre de substitutions (mots incorrectement reconnus effectivement), D le nombre de suppressions (mots omis à la reconnaissance) et I le nombre d'insertions (mots ajoutés à la transcription résultante) :

$$\begin{aligned} TauxMots &= 1 - WER = 1 - \frac{NbMotsMalReconnus}{NbMotsTotal} \\ &= 1 - \frac{S+D+I}{NbMotsTotal} = \frac{NbMotsTotal-S-D-I}{NbMotsTotal} \end{aligned} \tag{3.5}$$

Dans le cas des mathématiques, nous proposons de définir un autre taux relatif

uniquement aux mots ayant effectivement un sens du point de vue du langage des mathématiques, qu'on appellera par la suite **mots clés** (seuls les mots faisant référence aux symboles/relations mathématiques sont retenus en faisant abstraction de tous les mots de liaison et autres qui sont vides au sens du langage mathématique, voir la description associée au chapitre 7). Ce nouveau taux est défini à l'aide du nombre de mots clés non retrouvés $NbMotsClesMalReconnus$ (exprimé de la même façon que précédemment mais uniquement pour les mots clés) et du nombre de mots clés total $NbMotsClesTotal$. Ce nouveau taux est noté par $WER_{MotsCles}$. On en déduit alors le taux de reconnaissance $TauxMotsCles$ donnée par l'équation 3.6 :

$$TauxMotsCles = 1 - WER_{MotsCles} = 1 - \frac{NbMotsClesMalReconnus}{NbMotsClesTotal} \quad (3.6)$$

3.4 Bilan

Dans ce chapitre est présentée une autre approche pour l'édition et la manipulation de contenu mathématique en se basant sur la parole. Cette modalité offre une réelle alternative aux interfaces basées sur le clavier et la souris ou même à l'écrit. Toutefois, ces nouveaux modes d'interaction sont loin d'être aussi aboutis et matures que les autres. Dans les quelques travaux que nous avons pu identifier, il parait clair que l'apport que peut avoir une telle approche reste limité et ne peut intervenir réellement efficacement qu'en complément d'une autre modalité (l'écrit par exemple). Dans le chapitre suivant, nous présenterons un état de l'art sur le domaine de la combinaison de l'information pour pouvoir dégager les pistes à suivre dans la démarche de fusion des modalités audio et manuscrit en-ligne présentées jusqu'ici.

Chapitre 4

Fusion de données

Sommaire

C'est un état de l'art sur la notion de fusion qui est rapporté dans ce chapitre. Une présentation des niveaux et des techniques de fusion est faite, suivie par quelques applications recensées dans les travaux de la littérature.

4.1 Introduction

Aussi rapide et intelligente qu'elle soit, la machine demeure, néanmoins, moins perfectionnée que l'être humain. Cet ascendant qu'a l'homme sur sa machine est en partie dû à son raisonnement complexe dans la prise de décision. Les divers processus adoptés par l'homme dans sa réflexion pour résoudre des problèmes du quotidien ont inspiré énormément d'algorithmes. La vocation première de ces derniers est de rendre la machine de plus en plus autonome et intelligente, et donc de plus en plus efficace pour résoudre des problèmes complexes. C'est, par exemple, le cas des **réseaux de neurones artificiels** exploités à des fins de reconnaissance de formes. Ils sont largement inspirés du réseau neuronal biologique. Ils ont été proposés dès le milieu des années 40 par McCulloch et Pitts [MP43], perfectionnés, notamment par Widrow and Hoff, en 1960 puis par Rumelhalt et al. en 1962 et ne cesse d'être amélioré jusqu'à nos jours.

La **fusion de données** fait partie de ces domaines inspirés par le fonctionnement du cerveau humain. En effet, la combinaison d'informations est exploitée à différents niveaux par l'homme. C'est le cas notamment pour la distinction du relief (3D). Dans ce cas, grâce à sa vision stéréoscopique, l'être humain en voyant la même scène avec une légère différence de perspective entre les deux yeux, transmet à son cerveau (unité de calcul et de décision) les informations de profondeur nécessaires à la vision 3D. C'est également le cas pour l'estimation de la distance d'un objet grâce à l'utilisation conjointe de la vision et de l'ouïe. Sur la figure 4.1 est représenté un schéma résumant l'exploitation de ce processus de fusion chez l'homme. On y voit les différents capteurs naturels humains (les cinq sens), qui apportent diverses informations au centre de décision (le cerveau) en charge de combiner les flux d'information issus des cinq sens.

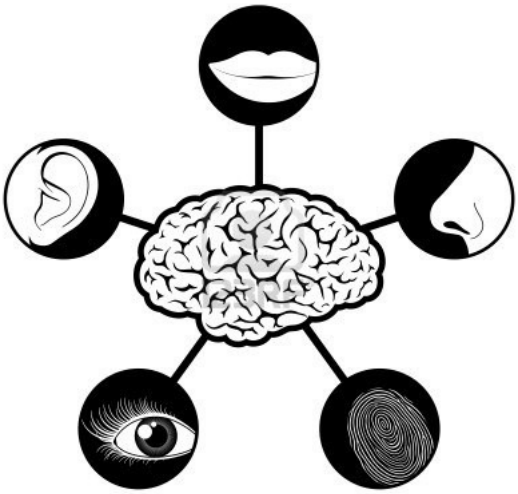


FIGURE 4.1 – Le cerveau et les cinq sens : mécanisme naturel de fusion chez l'homme [LeC]

En comparaison à d'autres domaines de recherche, la fusion de données reste un domaine relativement jeune quoique son usage semble assez naturel et intuitif [Rou94]. En effet, bien qu'on retrouve quelques rares travaux exploitant la combinaison d'information dans un cadre purement théorique dès la fin des années 60 [VT68, Dem68], ce n'est qu'au milieu des années 80 que les premières formulations algorithmiques dans ce domaine sont implémentées dans diverses disciplines [Gra06]. Dès lors, de nombreux travaux ont vu le jour, grâce notamment au **JDL** (U.S **J**oint **D**irectors **L**aboratories Data Fusion Group) qui a défini un modèle de fusion de données en décrivant de façon précise les fonctions qui s'y rapportent [Whi87, SBW99, LBR+04]. Ceci a permis de préciser certains points jusqu'alors mal défini concernant la fusion d'informations, en particulier : qu'entend-on par "fusion" ? Quels éléments fusionne-t-on ? Quels sont les niveaux de fusion ? Quelles en sont les étapes ? Dans la suite, nous allons donner la définition de ce concept de fusion de données tel qu'il a été proposé par le **JDL** et nous apporterons des éléments de réponse à ces questions.

4.2 Concept de fusion de données

De façon formelle, la *fusion de données* regroupe l'ensemble des techniques et algorithmes visant à combiner des informations multi-sources qui peuvent être aussi bien de même nature que de nature hétérogène. Ces informations peuvent être incertaines et/ou imprécises voire incomplètes. Le but étant de construire une nouvelle information qui en serait la combinaison et qui serait a priori plus fiable et moins ambigüe. Cette nouvelle information a pour vocation de faciliter le processus de prise de décision en exploitant la redondance et la complémentarité des sources fusionnées [LLH08].

Si le terme *fusion* de « *fusion de données* » est assez bien explicité dans la définition précédente, il n'en est pas de même du mot *données*. Dans ce contexte, par « *données* », on entend toute source d'information, qu'elle soit directement issue de mesures (brutes ou éventuellement pré-traitées), de décisions intermédiaires fournies par des experts ou d'une combinaison des deux formes précédentes [Fao00].

Il en découle que selon le type de données considérées, le niveau de la fusion change. C'est cette considération qui donne la première catégorisation des techniques de fusion en tenant compte du **niveau considéré**. Une autre façon de regrouper les techniques de fusion consiste à les distinguer suivant **les algorithmes et techniques mis en œuvre pour accomplir la tâche de combinaison**. Dans la suite nous donnons le détail de ces différents types de fusion.

4.3 Principales approches et différents niveaux de fusion

Comme indiqué à la fin de la section 4.2, on distinguera deux façons de catégoriser les techniques de fusion : soit par rapport aux algorithmes mis en œuvre pour accomplir cette tâche, on parlera alors des **approches de fusion**. Soit en consi-

dérant l'information mise en jeu dans le processus de fusion, auquel cas on parlera de **niveaux de fusion**. Commençons par exposer les différents niveaux de fusion, tout en sachant qu'à chacun d'entre eux s'appliquent les différentes techniques de combinaison que nous exposerons par la suite (section 4.3.2).

4.3.1 Différents niveaux de fusion de données

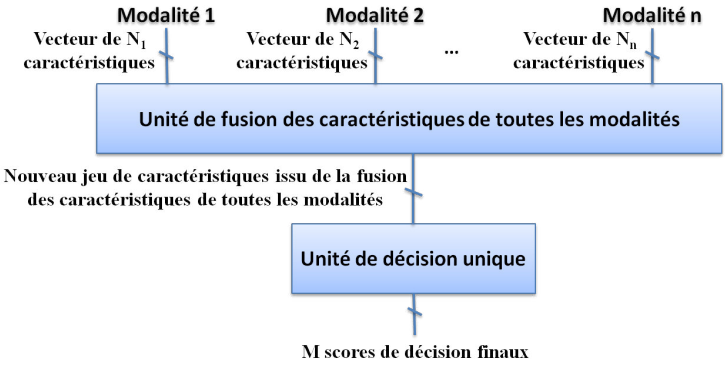
Identifier les différents niveaux de fusion revient à répondre à la question suivante : quels éléments fusionne-t-on ? Des éléments de réponse ont déjà été apportés dans la section 4.2. Pour plus de précision et en interrogeant la littérature sur cette question, les niveaux identifiés par le **JDL** dans leur premier modèle [Whi87] sont majoritairement ceux qui ont été retenus jusqu'à nos jours par les divers travaux publiés et qui se rapportent à cette discipline [BHA+01, Blo93, RF03, BAB+07, TJGD08, TMB09, AHESK10, KKKR13]. Globalement, deux principaux niveaux de fusion sont identifiés. Le premier groupe concerne la fusion dans l'espace des primitives (caractéristiques), appelée communément **fusion précoce**. Le second niveau concerne la fusion des décisions intermédiaires issues de systèmes experts amonts, on parle dans ce cas de **fusion tardive**. A coté de ces deux catégories, une troisième est définie. Cette dernière n'est rien d'autre que la combinaison des deux précédentes et elle porte la dénomination de **fusion hybride**. La figure 4.2 donne un aperçu général de ces différents niveaux. Dans la suite, nous allons nous intéresser plus en détail à chacun d'entre eux.

4.3.1.1 Fusion précoce

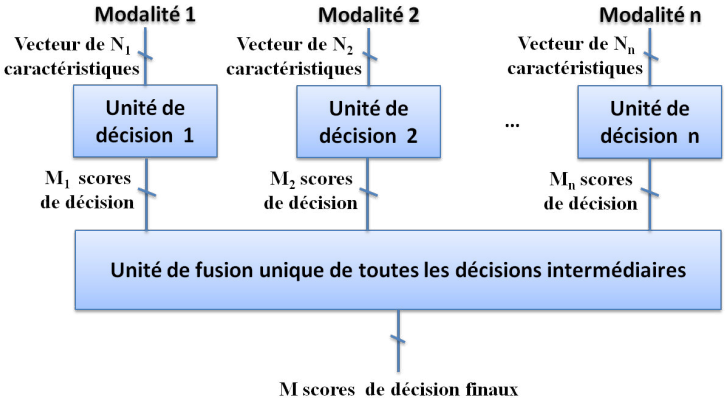
Dans ce cas on s'intéresse à la combinaison des primitives de chacune des modalités mises en cause dans le processus de fusion [Whi87, TMB09, AHESK10, KKKR13]. Il s'agit donc d'extraire les caractéristiques pour chacune des sources puis de les mettre dans un même format (normalisation à une échelle commune), à la suite de quoi, une seule unité de fusion est chargée de combiner l'ensemble des caractéristiques (figure 4.2a). La décision finale est prise par une seule unité qui reçoit en entrée l'information fusionnée. En opérant de la sorte, l'éventuelle corrélation qui existerait entre les différentes caractéristiques issues de chaque modalité est mise à profit. En revanche, cette façon de combiner les sources rencontre deux difficultés majeures qui peuvent mettre en échec son utilisation. Il s'agit de la finesse de la synchronisation requise des diverses modalités et de la normalisation indispensable des différents flux à la même échelle. Ce dernier point est loin d'être trivial [SWS05].

4.3.1.2 Fusion tardive

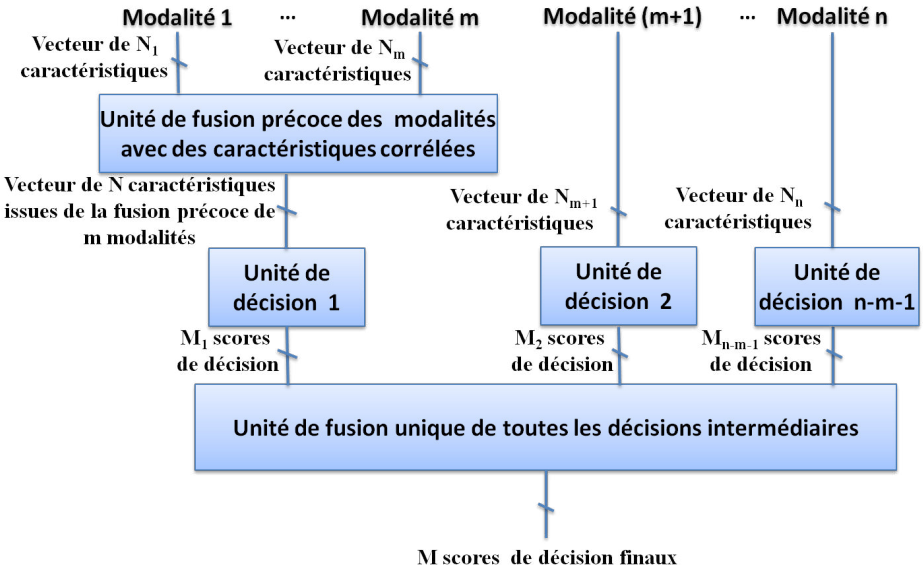
Dans cette approche, chaque modalité entrant dans le processus de fusion possède son propre système de décision spécialisé. En effet, dans ce cas, ce sont les décisions intermédiaires fournies par chacun des systèmes experts qui sont combinées. Dans cette configuration, il est toujours question d'une seule unité de fusion (qui combine les différentes décisions), mais d'autant de systèmes de décision qu'il y a de



(a) Fusion précoce



(b) Fusion tardive



(c) Fusion hybride

FIGURE 4.2 – Niveaux de fusion

sources à fusionner auxquelles se rajoute l'unité de décision finale (figure 4.2b). La fusion tardive met en exergue principalement deux avantages : le premier concerne la souplesse dans la normalisation des données combinées, notamment si les scores issus des systèmes experts sont probabilistes et bien adaptés ; le second point est lié au fait que le recours à des systèmes dédiés à chacune des modalités permet d'exploiter les méthodes d'analyse (reconnaissance) qui soient les plus adaptées à chacune d'entre elles. L'inconvénient majeur de ce type de système est la complexité de l'architecture (au moins autant de systèmes experts que de modalités).

4.3.1.3 Fusion hybride

Cette dernière façon de faire a pour but d'exploiter les points forts de chacun des niveaux précédemment décrits tout en limitant les inconvénients qui y sont induits. Tel qu'illustré par la figure 4.2c, dans cette configuration la fusion s'opère à la fois au niveau des primitives et au niveau des décisions intermédiaires. C'est ainsi que dans un cadre multi-modalités par exemple, les modalités qui présenteraient une corrélation dans le domaine des caractéristiques seraient précocement fusionnées. En revanche, celles qui présenteraient une certaine hétérogénéité ne seraient fusionnées qu'au niveau des décisions intermédiaires. Cela suppose donc que différents systèmes experts soient mis en œuvre ainsi que différentes unités de fusions.

4.3.2 Principales approches de fusion de données

Dans cette section, nous abordons les différentes façons de combiner les modalités selon les divers niveaux identifiés à la section 4.3.1 et donc la description du cœur de l'unité de fusion. La littérature regorge de méthodes exploitées à des fins de fusion de données. Néanmoins, que l'on se situe du point de vue du traitement de signal, de celui de l'image, de celui de la robotique ou d'une quelconque autre discipline, trois types de groupes de méthodes sont généralement identifiées [Gra06, AHESK10, KKKR13].

Les trois familles de méthodes sont : **les méthodes à base de règles, les méthodes basées sur des classifieurs et les méthodes à base d'estimateurs**. Dans la suite de cette section, nous développerons chacune d'entre elles. Nous présenterons pour chaque famille de méthodes celles que nous avons mis en œuvre dans le cadre des travaux de cette thèse. Ces méthodes font partie des plus courantes.

Avant de rentrer dans le détail de ces méthodes, nous allons commencer par présenter quelques notations sur lesquelles vont s'appuyer les formalismes qui vont suivre.

Soit $\mathbf{C}$ l'ensemble des $Nb_{Classes}$ classes de symboles possibles, défini comme suit $\mathbf{C} = \{C_1,\ C_2, \ldots, C_{Nb_{Classes}}\}$. Une hypothèse formulée au sein de la modalité i (parmi les N_M modalités à combiner), notée x_i, aura a être classée de telle sorte à prendre une des classes définies par $\mathbf{C}$. On définit également le score qu'une hypothèse soit de la classe C_j, par rapport à la modalité i, ou par rapport à la combinaison de toutes les modalités, par $S_i(C_j/x_i)$ et $S(C_j/x_1, \ldots, x_{N_M})$ respectivement.

À présent que ces définitions sont présentées, nous passons à décrire différentes méthodes de fusion et nous donnerons plus de détails sur certaines d'entre elles qui sont les plus utilisées. Ces dernières sont celles que nous avons mises en œuvre pour parachever la tâche de fusion dans notre système.

4.3.2.1 Fusion à base de règles

Dans ce cas la fusion s'opère moyennant une variété de règles de combinaison qui peuvent être statistiques (combinaison linéaire, maximum, minimum, vote majoritaire...), ou bien des règles particulières définies par l'utilisateur pour une application considérée [AHESK10]. Ce type d'outils est donc indépendant du contexte et ne prend en compte que les grandeurs fusionnées (caractéristiques, scores...). Dans cette catégorie de méthodes, deux sous-catégories peuvent être distinguées [Blo96] : celle des **opérateurs à comportement constant** et celle des **opérateurs à comportement variable**. La première sous-catégorie, comme son nom l'indique, regroupe tous les opérateurs qui se comportent de la même façon quelque soient les valeurs des grandeurs combinées. De façon formelle, si on définit un opérateur de fusion $\mathbf{F}$, et que l'on s'intéresse à fusionner, par son biais, deux grandeurs x et y prenant leurs valeurs dans I, alors cet opérateur ne peut suivre qu'**une seule** des **règles exclusives** données par les équations 4.1, 4.2 et 4.3, à la fois et ce quelque soient les valeurs des grandeurs x et y (comportement constant par rapport aux entités fusionnées) :

$$\forall (x,y) \in I^2, \mathbf{F}(x,y) \leq min(x,y), \tag{4.1}$$

$$\forall (x,y) \in I^2, \mathbf{F}(x,y) \geq max(x,y), \tag{4.2}$$

$$\forall (x,y) \in I^2, min(x,y) \leq \mathbf{F}(x,y) \leq max(x,y). \tag{4.3}$$

La règle du maximum, celle du minimum ou encore celle de la moyenne pondérée font partie de cette première sous-catégorie.

Concernant la seconde sous-catégorie, la règle de fusion varie en fonction des valeurs de x et y pour satisfaire une des conditions précédentes (équations 4.1, 4.2 et 4.3). Cela veut dire que l'opérateur $\mathbf{F}$ peut être dans certains cas pénalisant (satisfait l'équation 4.1). C'est par exemple le cas lors de la combinaison de scores et que le résultat après fusion est plus petit que le plus petit des scores initiaux (des modalités fusionnées). Dans certains cas, c'est une sur-évaluation qui est constatée (satisfaction de l'équation 4.2). Cette fois-ci, le score après fusion sera plus élevé que le plus élevé des scores donnés dans les modalités fusionnées. La méthode de rang de Borda, qu'on présentera par la suite [dB81] en est un bon exemple.

La fusion à base de règles permet d'avoir de bons résultats dès lors qu'un bon alignement des diverses modalités mises en jeu est assuré.

Quelques unes des méthodes de fusion les plus utilisées dans cette catégorie. Nous avons exploré six variantes de ces méthodes, comme nous allons le voir au cours du chapitre 7. 6. Trois d'entre elles sont des transformations linéaires par

le biais de moyennes arithmétiques avec des pondérations variables. Deux autres concernent des transformations non-linéaires en appliquant une pseudo-moyenne géométrique et la règle du maximum respectivement. La dernière de ces méthodes est la méthode de Borda explorée pour pallier le problème de normalisation des scores au sein des deux modalités considérées dans notre système.

Fusion à base de sommes pondérées [AHESK10]. Dans ce cas, pour le n-uplet d'hypothèses $(x_1, \ldots, x_{N_M})$, le score après fusion de chaque classe C_j est donné par la somme pondérée des scores de cette classe au niveau de chacune des N_M modalités. Elle est formulée par l'équation 4.4.

$$S(C_j/x_1, \ldots, x_{(N_M)}) = \sum_{i=1}^{N_M} w_{i,j} S_i(C_j/x_i), \tag{4.4}$$

où, $w_{i,j}$ est le poids associé à la classe C_j relativement à la modalité i. Le choix de ces poids obéit à la formule de normalisation $\sum_{i=1}^{N_M} w_{i,j} = 1$. Selon les valeurs de ces poids, on a défini trois types de sommes pondérées dans le cas de notre système. Celles-ci seront présentées à l'occasion du chapitre 7.

Fusion à base de produit [KHDM98]. Cette formulation est dérivée de la moyenne géométrique. En effet, prendre directement la moyenne géométrique pénalise les cas où les deux systèmes sont en parfaite opposition (l'un associe un score très fort à une hypothèse tandis que l'autre système lui attribue un score extrêmement faible). L'équation 4.5 rapporte la façon de construire le score à l'issue de la fusion.

$$S(C_j/x_1, \ldots, x_{(N_M)}) = 1 - \prod_{i=1}^{N_M} (1 - S_i(C_j/x_i)). \tag{4.5}$$

Fusion par la règle du maximum [RF03]. Dans ce cas, le score après fusion qu'un n-uplet d'hypothèses $(x_1, \ldots, x_{N_M})$ soit de la classe C_j est simplement le score maximal de ceux attribués aux niveau mono-modal $(S_1(C_j/x_1) \ldots S_{N_M}(C_j/x_{N_M}))$. Cela est représenté par la formule de l'équation 4.6.

$$S(C_j/x_1, \ldots, x_{(N_M)}) = \max_{i=1 \ldots N_M} (S_i(C_j/x_i)). \tag{4.6}$$

Fusion à base de la méthode de Borda [dB81]. Dans cette approche, l'idée est de pallier le problème de normalisation des scores. En effet, même si les scores au sein des différentes modalités sont dans l'intervalle $[0, 1]$, la dynamique au sein de cet intervalle pour ces modalités n'est pas la même. Ceci peut être handicapant pour les méthodes précédentes qui reposent sur les scores. Cette approche de comptage de Borda a pour objectif d'exploiter uniquement les rangs des classes et non leurs scores. Contrairement aux méthodes précédentes, le traitement ne se passe pas au niveau de chaque classe de façon isolée (connaissant son score), mais il est opéré sur l'ensemble de la liste des $N - meilleurs$ candidats. Cela, parce que

c'est l'information de rang qui est importante pour définir les nouveaux scores au niveau de chacune des deux modalités d'abord, puis, par conséquent le score après fusion.

Il existe énormément de variantes de cette méthode, et cela en fonction de la façon d'associer les votes pour chacune des classes.

4.3.2.2 Fusion basée sur des classifieurs

Comme son nom le suggère, cette méthode de fusion est basée sur des techniques de classification classiques [AHESK10]. Les classifieurs les plus populaires exploités à cette fin sont les machines à vecteurs supports (SVM), les réseaux de neurones (RN), les modèles bayésiens, ou d'autre basés sur la théorie des fonctions de croyance. Les unités de fusion embarquant ces techniques prennent en entrée des données multi-modales et fournissent en sortie soit les classes associées (rangement des classes selon la pertinence, solution finale) ou bien des caractéristiques fusionnées sur lesquelles se portera la décision finale par la suite.

Du point de vue de l'apprentissage artificiel, les méthodes de fusion par classification peuvent être regroupées en deux types :

Les modèles discriminatifs. Les *SVM* et les *RN* en sont de très bons exemples. Dans ce cas, on apprend les frontières séparant les différentes classes dans l'espace des caractéristiques multi-modales.

Les modèles génératifs . C'est le cas des modèles bayésiens par exemple, permettant à partir d'un modèle d'une classe d'exprimer à quel point une occurrence de caractéristiques peut être générée par ce modèle (en y associant un score ou une probabilité).

Nous avons dans ce manuscrit exploré et utilisé deux des techniques de cette catégorie. La première concerne l'utilisation des fonctions de croyance et la seconde le recours à un classifieur de type *SVM*. Nous donnons dans ce qui suit un aperçu de ces deux outils.

Les fonctions de croyance. Appelée également théorie de l'évidence, elle a été définie vers 1976 par Glenn Shafer [Sha76]. Elle est souvent identifiée également par la dénomination de la théorie de Dempster-Shafer du fait qu'elle repose sur les bases formulées, une dizaine d'années auparavant, par Dempster [Dem67]. Ce modèle, ainsi défini, se veut un bon outil pour la modélisation de l'incertain. Plus tard, dans son modèle des croyances transférable, Smets [Sme93, Sme98] fait la preuve que ce modèle de croyance est une réelle alternative aux probabilités subjectives [Van03]. Avant de présenter le mode opératoire de fusion par le biais de cette théorie, nous allons commencer par définir de façon brève le modèle du point de vue mathématique et donner les possibilités de combinaison qui sont offertes.

Présentation du modèle de croyance [Van03, PJME10]. Dans les travaux rapportés ici, on adopte le modèle de fonctions de croyance transférable (MCT) proposé par Smets et Kennes [SK94]. Ce modèle est basé sur deux idées de base :

la première concerne l'obtention de croyances, sur une question donnée, à partir de probabilités subjectives ; la seconde est la combinaison de ces degrés de croyance.

On ne va pas apporter ici tous les détails de ce modèle[1]. On va se focaliser sur les principaux éléments dont on aura besoin dans l'application qui est la notre, à savoir fusionner les informations issues des modalités manuscrite et audio.

De façon générale, cette théorie repose essentiellement sur un ensemble fini appelé *cadre de discernement* de l'expérience noté Ω et qui est défini par : $\Omega = \{\omega_1, \omega_2 \ldots \omega_N\}$. Il contient, en somme, toutes les propositions d'hypothèses de solution pour le problème considéré. La représentation de l'incertitude est faite par le biais de la notion de la fonction de croyance. Cette dernière est une fonction m définie de 2^Ω à $[0, 1]$ et satisfait la condition suivante :

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{4.7}$$

La quantité $m(A)$ représente la part de croyance exactement allouée à l'hypothèse A. Cette dernière étant un sous-ensemble de Ω. Les éléments A pour lesquels $m(A) > 0$ sont appelés *éléments focaux* de m. Dans le cas où l'ignorance est totale, $A = \Omega$ et $m(A) = m(\Omega) = 1$. Dans le cas où tous les éléments focaux A de m sont des singletons ($|A| = 1$), alors m tend à être une mesure de probabilité et est nommée dans ce cas *masse Bayésienne*.

Combiner deux masses de croyance m_1 et m_2, définies au sein du même cadre de discernement Ω, peut se faire à travers des opérateurs de combinaison variables. Dans ce travail, nous utilisons un des plus optimaux et des plus utilisés. Il s'agit de l'opérateur binaire conjonctif "$\cap$" [SK94]. La masse après combinaison $\widetilde{m}$ obtenue par cet opérateur est donnée par l'équation 4.8.

$$\forall A \subseteq \Omega, \widetilde{m}(A) = m_1(B) \cap m_2(C) = \sum_{B \cap C = A} m_1(B) m_2(C). \tag{4.8}$$

Il est, par la suite, possible de transformer une masse donnée m en la probabilité associée, très utile pour des fins de décision. Une distribution possible est nommée la probabilité *pignistique*. Le principe est de répartir de façon équitable la masse d'un sous-ensemble de Ω entre ses éléments. Cette distribution, notée P_m, est donc définie par Smets et Kennes [SK94] comme suit ($m(\emptyset) \neq 1$) :

$$\begin{cases} \forall \omega \in \Omega, P_m(\omega) = \sum_{A \subset \Omega} \dfrac{m(A)}{|A|(1 - m(\emptyset))} \delta_m(\omega), \\ \delta(\omega) = \begin{cases} 1 \ si \ \omega \in A, \\ 0 \ si \ \omega \notin A, \end{cases} \end{cases} \tag{4.9}$$

où $|A|$ est le cardinal de A.

1. Pour une présentation plus fine et détallée, consulter l'une de ces sources [Blo93, SK94, Sme98, Van03]

Le classifieur *SVM*. Les classifieurs en général et les *SVM* en particulier sont souvent utilisés pour combiner diverses modalités suivant divers modes, soit en prenant comme entrées la concaténation des vecteurs de caractéristiques issues de chacune des modalités fusionnées, soit en considérant comme caractéristiques d'entrée les décisions intermédiaires exprimées au niveau de chacune des modalités. Il est aussi possible de considérer des schémas de combinaison plus complexes (fusion hybride). L'objectif reste à chaque fois le même, réussir à trouver un nouvel espace de description suffisamment discriminant permettant d'apprendre les frontières entres les diverses classes du problème de façon plus précise que si l'on considérait uniquement le cadre mono-modal. Dans le cas du système que nous avons mis en place comme nous allons le voir au cours du chapitre 6, nous avons fait appel à un classifieur de type *SVM* pour fusionner les décisions intermédiaires fournies par les deux systèmes experts. Une brève description des SVMs et de leur apprentissage est donnée en annexe B.

4.3.2.3 Fusion basée sur les estimateurs

Cette approche de fusion a vu le jour à travers les applications liées à la poursuite d'objet dans un cadre multimodal. En effet, dans le cadre de l'**estimation** de la position d'une cible mouvante, cette dernière sera d'autant mieux localisée si plusieurs sources d'information sont mises en jeux (par exemple deux flux : les signaux audio et vidéo). Ceci est par essence la définition du processus de fusion de données.

Les méthodes couramment utilisées dans ce cas sont basées sur l'estimateur de Kalman, le filtre de Kalman étendu ainsi que les filtres particulaires.

Si ces trois exemples d'estimateurs couramment employés en fusion de données ont majoritairement pour cadre applicatif, comme indiqué plus haut, le suivi et la localisation de cibles, les points forts de chacun ne sont pas forcément les mêmes. En effet, le filtre de Kalman s'avère être très adapté lorsque le système suit un modèle linéaire. Le filtre de Kalman étendu quant à lui se comporte mieux quand les modèles sont non-linéaires mais Gaussiens. Enfin, la méthode reposant sur les filtres particulaires se veut être la plus robuste dès que les modèles sont non linéaires et non-Gaussiens [AHESK10].

4.4 Évaluation des systèmes basés sur la fusion

Dans le cas des systèmes multi-modaux, même s'il n'existe ni de mesure de performance standard, ni de cadre spécifique dédié à l'évaluation des algorithmes de fusion de données [KKKR13], différentes approches d'évaluation sont proposées dans divers travaux [HNS04, KGOI06, vL09, GJT13, KKKR13]. Dans [vL09], un état de l'art titré : *les challenges de l'évaluation, en pratique, des performances des systèmes de fusion d'information*, 52 articles scientifiques (conférences et journaux confondus) portant sur la problématique de fusion d'information ont été étudiés. Cette étude a montré que seulement 6% d'entre eux ont apporté une validation

pratique de l'architecture proposée. La stricte majorité restante se contente d'une évaluation par des simulations, parfois guidée par des mesures subjectives. Cela est majoritairement justifié par le manque de données vérités terrain. Globalement il existe deux catégories d'évaluation des systèmes multi-modaux [KÏ2, KKKR13]. La première est celle qui **évalue la qualité des données à fusionner**. Elle consiste à mesurer la fiabilité et la crédibilité des données en entrée du système de fusion à travers des mesures de corrélation entre-sources (combien de sources sont effectivement indispensables à la résolution du problème et le degré de conflit entre les sources fusionnées) [STA92, CBN04, Nim04]. La seconde catégorie d'évaluation des systèmes de fusion concerne **la mesure de performance du système lui-même**. En effet, dans ce cas, ce sont des critères d'évaluation du degré de succès ou de l'échec du système qui sont mis en œuvre. Dans ce cas, les métriques retenues sont très dépendantes du système et elles peuvent s'appliquer sur le système global ou sur chacun des sous-systèmes experts qui le composent. Ce qui est sûr, c'est que l'évaluation du système de fusion offre nettement plus de possibilités, à ce niveau là, qu'un système mono-modal [KKKR13]. Dans cette catégorie, on retrouve par exemple Wechsung et al, dans [WES+09], qui ont proposé une évaluation basée sur l'expérience utilisateur (*user experience* en anglais) en proposant, pour diverses tâches, de les accomplir en utilisant les modalités parole et geste, une fois dans un cadre mono-modal puis dans un cadre bi-modal les impliquant simultanément. Les utilisateurs sont à chaque fois invités à répondre à un questionnaire en donnant des notes subjectives. D'autres travaux rapportés notamment dans [Ovi03, RZD00] ont exploité le même procédé complété par des mesures objectives. Ces mesures objectives sont par exemple, dans le cas de [RZD00] où il est question d'accomplir des tâches de *CAD*[2], la fidélité du tracé résultat par rapport au modèle d'origine. C'est ainsi qu'il est demandé aux utilisateurs de recopier un objet se trouvant en fond d'écran par le moyen de différentes modalités (souris, stylo, parole). Différentes combinaisons de ces modalités sont explorées, y compris les cas mono-modaux. La pertinence du tracé réalisé est mesurée en faisant l'alignement des deux formes (vérité terrain et résultante). À coté de cela, le temps de saisi et de modification de l'objet sont mesurés pour chaque mode d'édition. Ces différents critères sont exploités pour comparer les différentes façons d'éditer l'objet. Dans [BD04, SN10], c'est l'expérience du magicien d'Oz qui est utilisée pour rendre compte de la qualité des systèmes multi-modaux mis en place. Un autre exemple est celui de Bernhaupt et al [BPWN07], qui ont utilisé un système de suivi du mouvement de l'œil (*eye-tracker* en anglais) pour l'évaluation du succès de la tâche de contrôle d'un satellite par un système multi-modal basé sur l'usage de deux manettes et de la parole de façon simultanée. Il existe énormément d'autres mesures de ce type, combinant approches subjectives et métriques objectives, qui sont définies pour des tâches bien spécifiques [CJM+97, KÖ3, MWS+09, TB11, BNPW07].

À présent, nous allons présenter de façon non-exhaustive quelques travaux majeurs que nous avons répertoriés sur la fusion de données. Pour ce faire, nous allons

2. Computer Aided Design (http ://en.wikipedia.org/wiki/Computer-aided_design)

respecter la catégorisation établie précédemment. Nous considérerons dans un premier temps les niveaux de fusion, et à chaque niveau des exemples associés aux différentes approches introduites plus haut sont donnés.

4.5 Quelques applications relatives à la combinaison d'informations

Comme nous l'avons stipulé en introduction de ce chapitre, depuis sa définition précise par le **JDL**, la fusion de données a connu un essor considérable et se retrouve dans des disciplines diverses et variées. Nous présentons dans la suite quelques-uns de ces travaux.

4.5.1 Travaux utilisant une fusion précoce

Dans la section 4.3.1.1, l'identification des pré-requis à une fusion précoce a été faite. Il est notamment question de disposer de sources présentant une homogénéité dans le domaine des caractéristiques, qui seraient synchronisées mais surtout qui seraient normalisées. Dans la suite nous en présentons quelques-uns.

4.5.1.1 Fusion à base de règles

Dans [FS02], Foresti et Snidaro ont exploité une fusion précoce à base de règles pour la détection et le suivi de personnes. Plus exactement, les positions $P(x(t), y(t))$ de la personne sur les trajectoires fournies par différents types de capteurs (radar, caméra infra rouge, caméra optique) sont fusionnées localement par le biais d'une combinaison linéaire donnée par l'équation 4.10. Dans cette équation, à chacune des positions issues de chaque capteur (modalité), désignée par $P_i(x(t), y(t))$ (i représentant la modalité considérée parmi les M modalités possibles), à un instant donné est associé un poids w_i exprimant la contribution de la modalité en question dans l'expression de la position finale.

$$P(x(t), y(t)) = \begin{cases} x(t) = \dfrac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i \times x_i(t) \\ y(t) = \dfrac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i \times y_i(t) \end{cases} \tag{4.10}$$

Wang et al. quant à eux ont rapporté dans [WKYJ03] des travaux sur le suivi en temps réel de piétons dans des séquences de vidéo surveillance. Ils ont également proposé une approche pour fusionner les différentes vidéos (issues des différentes caméras) et n'afficher sur un moniteur unique que la vue présentant un intérêt pour la personne en charge du contrôle à un instant donné (application de la règle du maximum de saillance). Ceci allège la tâche de l'agent en charge du contrôle des différentes scènes à surveiller.

De leur coté, Chang et al. [CBSV03] ont exploré la fusion précoce dans le cadre d'un problème d'authentification par le visage et l'oreille. La méthode utilisée consiste à concaténer deux images. La première est celle du visage et l'autre est celle de l'oreille correspondante avant d'en extraire un vecteur de caractéristiques, sur lesquelles est faite l'authentification.

4.5.1.2 Fusion basée sur des classifieurs

La fusion précoce à base de classifieurs est également explorée dans divers domaines. C'est le cas par exemple de Pitsikalis et al. qui ont fusionné des caractéristiques audio (*coefficients MFCC* [GME11]) et des caractéristiques vidéo (*descripteurs de forme et de texture*) à travers un modèle Bayésien (*cf.* équation 4.11) à des fins de reconnaissance audio-visuelle de la parole [PKPM06].

$$p(C|f_1, f_2 \dots f_M) = \frac{1}{N} \prod_{i=1}^{M} p(f_i|C)^{w_i} \tag{4.11}$$

L'équation 4.11 représente de façon générale la fusion de caractéristiques issues de M modalités. Un vecteur f_i de caractéristiques est extrait de chaque modalité i. Le résultat de la fusion est la classe $\hat{C}$ choisie dans l'ensemble $\sum_c$ des N_c classes possibles. La classe $\hat{C}$ résultante est celle qui satisfait la règle du **maximum de probabilité a postériori** ($\hat{C} = argmax_{C \in \sum_c} p(C|f_1, f_2 \dots f_M)$). La quantité w_i quant à elle représente le poids alloué à la modalité i. Elle exprime la contribution de cette modalité dans la décision finale de la classe qui est retenue. Ces poids doivent satisfaire la condition de normalisation suivante : $\sum_{i=1}^{M} w_i = 1$. La constante N est un facteur de normalisation de la probabilité a posteriori $p(C|f_1, f_2 \dots f_M)$.

Dans un autre travail, rapporté par Mena et Malpica [MM05], une fusion au niveau des caractéristiques a été réalisée à l'aide de la théorie de *Dempster-Shafer* (*cf.* section 4.3.2.2) pour la segmentation d'images satellitaires a été proposée. En effet, en partant de trois mesures au niveau pixel (dénotées respectivement comme étant des statistiques d'ordre un, car ne considère que le pixel ; d'ordre un et demi, car le voisinage du pixel est pris en compte ; et enfin d'ordre deux considérant non seulement le voisinage mais aussi les trois bandes de l'image *RVB*), qui peuvent chacune être utilisées de façon individuelle pour accomplir cette tâche de segmentation. Selon [MM05], ces trois sources d'information présentent des inconvénients de type différent mais surtout ont l'avantage d'être complémentaires. C'est ainsi que la solution proposée améliore grandement les résultats de segmentation par rapport à une seule modalité (utilisant une seule des trois mesures).

Les réseaux de neurones ont également été explorés à des fins de fusion de caractéristiques. Dans [CD00], un réseau de neurones à convolution (*TDNN : Time Delay Neural Network*) est utilisé pour exploiter la corrélation existante entre le signal de la parole et le mouvement des lèvres pour faire de la détection et de la localisation de locuteur. Le mode opératoire de ce *TDNN* est de prendre en entrée un vecteur formé par la concaténation de deux jeux de caractéristiques. Le premier jeu de caractéristiques est issu de la modalité audio et correspond aux coefficients

MFCC [GME11]. Le second quant à lui est donné par le flux vidéo sous la forme des déplacements observés sur la région des lèvres entre deux images successives. La sortie de ce réseau est un neurone unique indiquant si un locuteur est détecté ou pas dans la zone et au temps considérés.

Wang et al. ont utilisé une approche de fusion de descripteurs audio (incluant la détection de l'activité du signal, la bande de fréquence, la fréquence fondamentale...) et vidéo (histogrammes de couleur, cartes de déplacement, détection de formes...) pour de la classification de scènes de films [WLH00]. Ces vecteurs sont concaténés et sont utilisés comme un jeu de caractéristiques unique en entrée d'un classifieur de type Modèle de Markov Caché (*MMC*). À l'occasion de cette étude, il a été clairement démontré que le flux audio a un apport capital quant au bon déroulement de cette tâche de catégorisation.

Des outils de la théorie de l'information sont également mis au service de la fusion de données. C'est le cas notamment du **modèle d'entropie maximale** donné par l'équation 4.12. Dans [MaR10], Magalhães et Rüger ont exploité ce modèle pour combiner des informations textuelles et visuelles afin d'accomplir une tâche d'indexation sémantique d'images.

$$P(C|f_i, f_j) = \frac{1}{N(f_i, f_j)} e^{F(f_i, f_j)}. \tag{4.12}$$

Dans l'équation 4.12, f_i *et* f_j représentent respectivement les vecteurs de caractéristiques des modalités i (*texte*), j (*image*). L'opérateur F assure la combinaison des vecteurs de caractéristiques f_i *et* f_j. le paramètre $N(f_i, f_j)$ quant à lui est un facteur de normalisation dépendant de f_i *et* f_j assurant que le résultat de combinaison ($P(C|f_i, f_j)$) soit homogène à une probabilité que ce jeu de caractéristiques (f_i *et* f_j) soit associé à la classe C.

4.5.1.3 Fusion à base d'estimateurs

La fusion à base d'estimateurs, comme nous l'avons vu dans la section 4.3.2.3, est très adaptée à la fusion précoce d'information. On la retrouve dans des problématiques relatives à l'estimation de diverses grandeurs (position, vélocité, accélération, trajectoire...). Cela est essentiellement dû à la méthodologie suivi par ce type de méthodes ainsi que les applications qui étaient à son origine.

C'est ainsi que Potamitis et al. [PCT04] ont utilisé le filtre de Kalman (**FK**) [Kal60] pour estimer la position d'un individu évoluant dans un environnement au sein duquel plusieurs autres individus sont en mouvement. Chacun d'entre eux est suivi par un réseau de capteurs audio (microphones). Un estimateur de type **FK** est dédié à chaque individu. Le modèle adopté dans ce travail est régi par le système d'équations 4.13 :

$$\begin{cases} \mathbf{s}_i(t) = \mathbf{F}\mathbf{s}_i(t-1) + \mathbf{v}_i(t), \\ \\ \mathbf{y}_i(t) = \mathbf{H}\mathbf{s}_i(t) + \mathbf{w}_i(t). \end{cases} \tag{4.13}$$

Dans ce système d'équations on retrouve le caractère récursif caractéristique du *filtre de Kalman*. Cela revient à dire que pour estimer l'état courant $\mathbf{s}_i(t)$ et l'observation actuelle $\mathbf{y}_i(t)$ du capteur i, à l'instant t, on a uniquement besoin de l'état du système à l'instant $t-1$ ($\mathbf{s}_i(t-1)$). Cette formulation admet également que le modèle est linéaire et entaché de bruit blanc Gaussien. La matrice $\mathbf{F}$ représente le modèle de transition, tandis que $\mathbf{H}$ est nommé modèle d'observation. Les grandeurs $\mathbf{v}_i(t)$ et $\mathbf{w}_i(t)$ sont respectivement des bruits qui entachent l'état et l'observation pour le capteur i. Ces deux bruits obéissent à des lois normales à moyennes nulles ayant respectivement des matrices de covariance $\mathbf{Q}_i$ et $\mathbf{R}_i$. Du point de vue pratique, le vecteur d'état à l'instant t est donné en fonction des position (x, y, z) comme suit : $\mathbf{s}(t) = [x(t)\ \dot{x}(t)\ y(t)\ \dot{y}(t\ )z(t)\ \dot{z}(t)]$. Le vecteur d'observation pour sa part est formulé comme étant : $\mathbf{y}(t) = [x(t)\ y(t)\ z(t)]$. Les éléments des matrices $\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}_i$ et $\mathbf{R}_i$ sont fixées en considérant l'hypothèse de mouvement Newtonien.

Les filtres particulaires sont d'un usage très répandu dans le domaine de l'analyse de contenu multimédia dès qu'il est question de signaux séquentiels suivant des modèles non-linéaires et non-Gaussiens [DKZ+03]. Ils sont également connus sous la dénomination de méthode de *Monte Carlo séquentielle*. Dans [VGBP01], la fusion à base de filtres particulaires est exploitée pour mieux suivre un locuteur en utilisant à la fois l'information vidéo, issue d'une caméra, ainsi que le flux audio de deux microphones. Les particules dans ce cas sont les caractéristiques audio-visuelles.

4.5.2 Travaux utilisant une fusion tardive

Bien souvent, la fusion concerne des modalités hétérogènes et la normalisation des caractéristiques est peu évidente (*cf.* section 4.3.1). C'est pourquoi la littérature est abondante d'applications exploitant une fusion tardive qui se veut un moyen de s'affranchir de cet obstacle d'hétérogénéité, du moins y être moins sensible.

4.5.2.1 Fusion à base de règles

Dans [NMS+00], Neti et al. ont présenté les résultats de fusion de décisions de deux systèmes experts en charge des modalités audio et vidéo pour l'identification du locuteur. En effet, les listes des $N meilleurs$ scores de classes données par les deux systèmes amonts (audio et vidéo) sont combinées à travers une simple somme pondérée comme le montre l'équation 4.14.

$$d_{f,i} = w_a d_{a,i} + w_v d_{v,i}, \tag{4.14}$$

où $d_{f,i}$, $d_{a,i}$, $d_{v,i}$ sont respectivement les scores de décision après fusion, du classifieur audio et du classifieur vidéo pour le locuteur i. Et w_a et w_v sont les poids assignés à chacune des deux modalités audio et vidéo.

Pour obtenir l'identification du locuteur à partir d'une source audio, Radova et Psutka ont eu recours à la règle du vote majoritaire. Le signal audio est, dans un premier temps, découpé en segments élémentaires correspondant à un certain nombre de voyelles. Chaque segment est classé grâce à deux types de classifieur : le

premier opérant dans le domaine temporel est basé sur l'algorithme de déformation temporelle dynamique (DTW). Le second, quant à lui, est un classifieur *du plus proche voisin (1ppv)* qui opère dans le domaine fréquentiel en considérant les sept premiers coefficients cepstraux du type LPC. Ensuite un vote majoritaire est opéré sur l'ensemble de ces décisions intermédiaires sur tous les segments et pour les deux types de classifieurs [RP97].

D'autres travaux sur la fusion à base de règles ont proposé des définitions de règles spécifiques au problème abordé.

Par exemple, Pfleger [Pfl04, Pfl05] a proposé une architecture de fusion qui s'appuie sur des règles de production à base de pondération et de structures conditionnelles. Ces règles sont appliquées sur les décisions intermédiaires issues de deux systèmes amonts qui sont en charge des modalités manuscrit en-ligne (stylo numérique pour être exacte) et audio. Dans cette approche, la sortie du module de fusion est soit une copie de l'un des deux systèmes experts (grâce aux structures conditionnelles), soit une combinaison des deux experts à travers les pondérations définies (combinaison linéaire pondérée).

4.5.2.2 Fusion à base de classifieurs

Le recours à des méthodes basées sur de l'apprentissage s'est avéré, au fil des travaux rencontrés, être une des meilleures si ce n'est la meilleure approche dès qu'il s'agit de définir les pondérations adéquates à chacune des décisions intermédiaires données par les modalités impliquées dans le processus de fusion. Nous rapportons dans la suite quelques-uns de ces travaux.

Un exemple de fusion de données tardive par le biais d'un classifieur bayésien est rapporté dans le travail de Meyer et al. dans [MMW04] pour de la reconnaissance bimodale de chiffres. En effet, deux classifieurs de type HMM se chargent d'effectuer la classification des signaux audio (à partir des caractéristiques $MFCC$) et visuel (en extrayant les contours des lèvres) de façon indépendante. Ensuite, les deux listes des $N - meilleures$ solutions (avec leurs scores sous forme de probabilités) sont combinées par le biais du classifieur bayésien.

Dans une autre application tirée du domaine de l'identification et de la vérification d'empreinte digitale, Singh et al. ont combiné les scores de quatre types d'algorithmes de classification d'empreintes grâce à la théorie de D-S [SVNS06]. Dans ce travail, les scores intermédiaires sont issus de :

1. la classification des points caractéristiques (*minuties-based matching*) qui se révèle être une très bonne approche dans le cas d'images avec de bonnes résolutions mais qui est facilement mise en échec dès qu'il est question d'images de moins bonne qualité ;
2. l'alignement des squelettes des empreintes en considérant les arêtes comme caractéristiques (*ridge-based matching*). Cette méthode a de moins bonnes performances que la précédente dans le cas d'images de bonne résolution mais se comporte mieux vis-à-vis des images de faible résolution ;
3. l'alignement des codes associés à l'empreinte (*Fingercode-based matching*) est

réalisé. Ce code est issu de l'application d'un banc de filtre de type **Gabor** donnant une vision à la fois globale et locale de l'empreinte (analyse multi-résolution) ;

4. l'alignement par le biais de l'algorithme de **RANSAC** des pores des empreintes (*pores-based matching*), connus pour être très discriminatoires mais en même temps très difficiles à extraire.

Les différents scores sont combinés par l'opérateur de Dempster donné dans l'équation 4.8. Différentes combinaisons de fusion des classifieurs précédents ont été explorées et l'approche mettant en jeu les quatre classifieurs s'est avérée être de loin la plus pertinente pour ce problème. L'adoption de la théorie de *D-S* est confrontée à d'autres approches de fusion de la littérature existantes dans le domaine et a prouvé sa supériorité.

Un classifieur de type réseau de neurones est employé pour fusionner les scores de deux systèmes experts en charge de mesurer des grandeurs (“ modalités ”) complètement hétérogènes [GMS+03]. La première reflète l'état de l'activité sur le réseau informatique d'un laboratoire suivant différents critères (le nombre d'authentifications et le taux d'usage du réseau et des ressources). La seconde quant à elle, est l'observation de l'activité physique des personnes en déplacement grâce à un réseau de caméras *CCD*. L'unité de fusion, qui est le *RN*, combine donc les scores des deux types de mesure afin de se prononcer sur l'activité (degré de l'activité) humaine dans l'enceinte du laboratoire.

Les classifieurs de type Support Vector Machine sont parmi ceux qui sont les plus usités en fusion de décisions. Énormément de travaux reportent un tel usage dans la littérature. C'est notamment le cas par exemple de Bredin et Chollet dans leur article sur l'identification biométrique de tête parlante [BC07]. Le travail rapporté concerne l'identification de visages dans des vidéos. Dans un premier temps, une identification par la parole et une identification faciale par les images sont réalisées dans le cadre mono-modal. Par la suite, les scores issus de ces deux systèmes amonts sont pris en entrées d'un classifieur *SVM* en charge de les fusionner et de se prononcer sur l'identité définitive de la personne.

4.5.3 Travaux utilisant une fusion hybride

Bien que, théoriquement, l'intérêt de la méthode hybride soit de tirer partie des deux autres méthodes (précoce et tardive), peu de travaux sont à signaler jusqu'à présent. En voici quelques-uns.

4.5.3.1 Fusion à base de classifieurs

Dans [IR10], une fusion hybride est utilisée pour proposer un système d'identification du locuteur basée sur le contenu vidéo (information audio-visuelle). Deux types de caractéristiques audio (*MFCC* et *LPCC*) sont considérées et combinées par le biais de *HMM*. Le même mode opératoire est appliqué pour la partie visuelle, où des caractéristiques décrivant l'émotion et la forme du visage sont également

combinées à travers un classifieur du type *HMM*. Les décisions de chacun des systèmes associées à d'autre mesures, reflétant la fiabilité des deux *HMM* experts, sont par la suite fusionnées pour rendre la décision finale. La fusion est accomplie grâce à un autre classifieur du type *HMM* dont les caractéristiques sont issues de la concaténation de tous les scores précédents.

Zhu et al., quant à eux, ont exploité un classifieur de type *SVM* pour mettre en œuvre un système de catégorisation d'images contenant du texte [ZYC06]. Dans ce travail, deux étapes principales se succèdent : la première consiste à associer des probabilités d'appartenance de l'image à chacune des catégories possibles en combinant différentes caractéristiques visuelles. La seconde, faite parallèlement à la première, consiste à détecter le texte des images. À partir de ces zones de texte, des caractéristiques sont extraites. Une fois ces deux types d'information disponibles, un vecteur de caractéristique es formé par la concaténation des probabilités d'appartenance issues de l'image et des descripteurs des zones de texte. Ce vecteur est utilisé comme entrée d'un classifieur de type *SVM* qui est en charge de fournir la décision finale concernant la catégorie de l'image.

4.6 Bilan

Durant ce chapitre, la notion de **fusion de données** a été abordée. Cette dernière est au centre de ces travaux de thèse. Les différentes notions que nous allons aborder par la suite ont été présentées.

Du point de vue de notre problématique, la combinaison des flux audio et manuscrit en-ligne, énormément de possibilités se dégagent à partir de ce travail bibliographique sur la notion de combinaison d'informations. Nous reviendrons plus en détail sur cela pour faire le tri et justifier les choix que nous avions à prendre durant cette thèse (au cours du chapitre 6 en particulier).

De façon générale, il est nécessaire de préciser certaines précautions d'usage liées à la fusion de données.

1 - **Choix de ce qui est fusionné :** que l'on soit dans une configuration où les modalités sont homogènes ou non, le choix de fusionner les caractéristiques ou les décisions, voire les deux est très important quant au succès ou pas du processus de fusion.

2 - **Synchronisation des modalités :** en fonction de ce qui est fusionné, l'alignement des modalités s'avère être plus ou moins indispensable. À titre d'exemple, fusionner les caractéristiques requiert une synchronisation très fine des différentes modalités en cause. En revanche, fusionner les décisions demande moins de finesse.

3 - **Normalisation de données :** quelque soit le type de données fusionnées, leur normalisation est impérative. Si elle n'est pas assurée, la prise en compte des diverses sources est biaisée. Cette tâche de normalisation est moins ardue dans le cadre de fusion de décisions.

4 - **Sélection de modalité(s) :** c'est en particulier très utile quand plusieurs

modalités (plus de deux) sont en présence. Souvent, selon l'instance à traiter à un instant t, certaines des modalités considérées sont suffisantes pour une bonne prise de décision. Il peut même arriver, dans certains cas, que considérer l'ensemble des flux est nuisible au processus de décision.

5 - **Réduction de la taille des vecteurs de données fusionnées :** il peut arriver, dans le cas de fusion de caractéristiques notamment, que la taille de l'espace de représentation des données requise dans le cas mono-modal, ne soit pas nécessaire dans le cas multi-modal. Ne pas la réduire induira une plus grande complexité du modèle de fusion que ce que requiert le problème abordé.

Deuxième partie

Contributions de la thèse

Chapitre 5

HAMEX, une base bi-modale d'expressions mathématiques

Sommaire

La base bi-modale d'expressions mathématique *HAMEX* est présentée au lecteur au cours de ce chapitre. Ce dernier retrace l'ensemble du processus de construction de cette base depuis la constitution du corpus qui la compose à la version collectée annotée des données en passant par les protocoles de collecte/annotation.

5.1 Introduction

Dans la mesure où l'objectif de ces travaux est de mettre en œuvre un système de reconnaissance d'expressions mathématiques disponibles en format bi-modal (tracé manuscrit en-ligne et parole), il est nécessaire de disposer de données disponibles sous ces deux formats.

Nous l'avons introduit au cours des chapitres précédents, aussi bien la communauté de la reconnaissance de l'écriture manuscrite que celle de la transcription automatique de la parole, ont exprimé un intérêt grandissant envers la problématique d'interprétation des formules mathématiques. La maturité des systèmes proposés, dans l'une ou l'autre des deux modalités, n'est pas la même. En effet, tel qu'on l'a rapporté à l'issue du travail bibliographique, les solutions apportées dans le cas de l'utilisation du signal manuscrit sont plus abouties.

Quelque soit la modalité, la mise à disposition de bases de données, nécessaires à la fois durant la phase de réalisation du système et celle de sa validation, est cruciale. Ces bases de données doivent contenir, non seulement le **signal brut** lui-même, mais aussi tout un ensemble d'informations permettant d'identifier de façon unique et non-ambigüe l'EM représentée par le signal : la **vérité terrain**.

À notre connaissance et au moment de la rédaction de ce manuscrit, on ne répertorie aucune base de données d'expressions mathématiques parlées. Du moins, qui soit disponible publiquement voire simplement publiée. Les quelques systèmes que nous avons cités à la section 3.2.2, ne mentionnent pas de bases dédiées aux mathématiques. Ceci est dû au fait que les ressources (modèles acoustiques, dictionnaire phonétisé et modèles de langages, section 3.2.1.3) sont privatives. De plus, l'apprentissage du modèle acoustique requiert un corpus de paroles transcrites quelconque et pas forcément dédié aux mathématiques. Le dictionnaire phonétisé quant à lui est intimement lié au modèle acoustique et il est défini par un expert. Ces deux premières ressources sont fournies par les systèmes de transcription de la parole utilisés qui sont, dans le cas de toutes les applications citées dans cette thèse, commerciaux. Enfin, le modèle de langage qui lui désigne le champ d'application visé par le système : le langage mathématique ici, est également défini en considérant le domaine visé par l'application mise en place.

Du coté de la modalité manuscrite en-ligne, ainsi que nous l'avons également rapporté dans la première partie de ce rapport, les méthodes sont plus nombreuses et même quelques systèmes de démonstration en-ligne sont disponibles. Toutefois, jusqu'à un passé très récent, il n'existait pas de base de données commune qui permettrait de comparer de façon objective ces méthodes entre elles. C'est pour remédier à cette lacune que Mouchère et al. [MVGG+11, MVGK+12, MVGZ+13] ont mis en place une compétition sur la reconnaissance d'expressions mathématiques manuscrites en-ligne (*CROHME*, l'acronyme anglophone pour : Competition on Recognition of Online Handwritten Mathematical Expressions). Sur ses différentes éditions, les organisateurs de cette compétition ont mis à disposition des bases de données complètement annotées, aussi bien en apprentissage/validation qu'en test. De plus, tous les participants sont classés suivant les mêmes métriques

d'évaluation appliquées sur les mêmes corpus.

Des bases bi-modales, fournissant à la fois la version manuscrite en-ligne et la version parlée de chacune des EMs sont, à notre connaissance, inexistantes. Pour les besoins des travaux de cette thèse, nous avons collecté une base bi-modale d'EMs baptisée *HAMEX* [QMS+11]. Dans les sections qui vont suivre, nous allons présenter *HAMEX*, le mode opératoire de sa construction, sa phase de collection, et enfin sa phase d'annotation.

5.2 Construction du corpus constituant *HAMEX*

HAMEX est le nom de baptême de notre base bi-modale d'expressions mathématiques. Ce nom est tiré de la description anglo-saxonne de la base, à savoir : **H***andwritten and* **A***udio dataset of* **M***athematical* **EX***pressions*. *HAMEX* est composée de 4350 expressions mathématiques différentes, disponibles chacune sous forme manuscrite et sous forme audio. La longueur moyenne (en symbole) des EMs est de 14.5, allant de 4 symboles pour la plus courte à 50 pour la plus longue. Le corpus inclut 74 classes différentes de symboles. *HAMEX* a été définie de façon à couvrir de nombreux domaines, puisqu'elle est générée à partir d'un corpus qui se veut le plus réaliste possible. En effet, au lieu de procéder comme pour d'autres bases qui sont issues de grammaires génératrices de leur corpus, à l'image de la base *MathBrush* [LMM+06], dans le cas de *HAMEX* c'est un corpus extrait de textes, de divers disciplines, contenant des formules mathématiques. On reviendra sur cela dans la suite.

La constitution du corpus d'expressions mathématiques est une étape cruciale et délicate. En effet, il est important de le construire en respectant certains critères. Il faut que le corpus soit :

1. **le plus réaliste possible :** la disposition des EMs et l'ordre des symboles ne peuvent être arbitraires : ils doivent respecter des règles de grammaire et couvrir un domaine d'application.
2. **le plus général possible :** il est important de disposer d'une variabilité des domaines considérés. Cela veut dire que la base doit couvrir des domaines aussi variables que la biologie, les statistiques, l'électronique, la physique...
3. **le plus riche possible :** en terme de complexité des EMs (longueur, relations spatiales impliquées) afin de pouvoir évaluer les systèmes avec des degrés de difficultés différents.
4. **le plus représentatif possible du point de vue du vocabulaire :** face au grand nombre de symboles mathématiques existants, il est intéressant d'inclure les plus fréquents du monde réel.

Pour satisfaire ces objectifs, nous nous sommes appuyés sur deux méthodologies complémentaires. D'une part, nous avons conçu une partie du corpus en nous appuyant sur une grammaire générative permettant de fabriquer des expressions simples et maitrisées, de type calculette, et d'autre part, nous avons extrait le reste du corpus de sources réelles disponibles sur la toile. Dans ce cadre, nous avons extrait

toutes les EMs présentes dans les pages composant l'encyclopédie libre en-ligne *Wikipédia* dans sa version française. Cette ressource regorge de documents de tous les domaines librement accessibles, offrant un volume immense d'EMs. Cette première extraction a permis d'obtenir plus de 75 000 expressions, correspondant à plus de 1 100 000 symboles. Le nombre de classes de symboles différentes était de plus de 600. Ces EMs ont par la suite été filtrées en supprimant les EMs ayant un seul symbole (inséré dans le texte). Les EMs excessivement longues ont aussi été enlevées du corpus.

Pour compléter cette partie du corpus, nous avons comme indiqué précédemment généré des EMs simples composées de peu de symboles (chiffres et opérateurs de base : addition, soustraction, multiplication, division, égalité, inégalité) et de relations simples (de type paire horizontale). Ces dernières sont, dans l'ensemble, des formules de comparaison et d'arithmétique. Elles sont générées de façon aléatoire, grâce au codage de Prüfer [AH78]. Ce dernier permet, à partir d'une séquence de $n-2$ entiers, de générer un arbre étiqueté à n nœuds. Chacun des entiers représente le label du nœud. L'ordre dans lequel ces entiers apparaissent dans la séquence permet de déduire les connexions au sein de l'arbre. Une fois l'arbre disponible, un arbre de dérivation d'une EM est déduit. Le passage à cet arbre donnant l'EM est obtenu en substituant, de façon aléatoire, les nœuds terminaux par des nombres. Les autres nœuds quant à eux sont remplacés, de façon aléatoire également, par divers opérateurs binaires. Sur la figure 5.1 est donné un exemple d'obtention de l'arbre de dérivation de l'EM à partir du code de Prüfer, en passant par l'arbre binaire étiqueté.

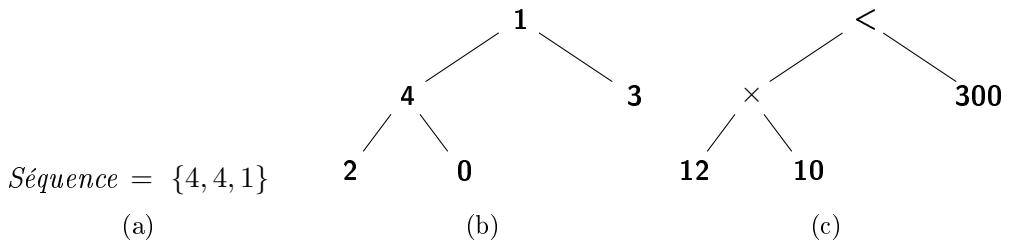


FIGURE 5.1 – Exemple de génération d'une EM ($12 \times 10 < 300$) à partir du Codage de Prüfer. (a) Séquence correspondant au code de Prüfer ($\{4, 4, 1\}$). (b) Arbre binaire étiqueté déduit. (c) Arbre de dérivation de l'EM

Finalement, la base *HAMEX* peut être découpée en trois sous-corpus de complexité variable. Le premier, le plus simple, représente toutes les équations de type calculatrice, incluant les dix chiffres et quelques opérateurs binaires ainsi que les parenthèses. Elles sont générées à partir du codage de Prüfer exposé précédemment. Les deux autres sont composés des expressions extraites de *Wikipédia*, qui sont organisées en deux catégories, de complexité croissante. Ces deux sous-corpus ont un nombre de classes différent et font intervenir des complexités au niveau relationnel variables.

Dans le tableau 5.1, est résumé le découpage de la base *HAMEX* en ses trois

corpus, en reprenant le nombre et le type des classes de symboles qui sont impliquées dans chaque sous-corpus ainsi que le nombre d'EMs incluses.

Corpus		Calculette	WIKIEM	WIKIEM-EXT
Taille du vocabulaire		25	56	74
Nombre d'EMS		870	1 740	1740
Nombre de symboles		17 478	17 020	21 390
Type de symboles	**Lettres latines (minuscules + majuscules)**		$abcdefgikn$ $rsxyz\ XY$	$a-z\ XY$
	Lettres grecques		$\alpha\beta\gamma\phi\pi\theta$	$\alpha\beta\gamma\phi\pi\theta$
	Chiffres	$0-9$	$0-9$	$0-9$
	Opérateurs binaires	$+-\pm\times/\div$ $=\neq<\leq>\geq$	$+-\pm\times/\div$ $=\neq<\leq>\geq$	$+-\pm\times/\div$ $=\neq<\leq>\geq$
	Opérateurs ensemblistes			$\in\ \forall\exists$
	Opérateurs élastiques	()	$()\ \sum\ \int$ $\sqrt{\ }\ \overline{\ }$	$()\ \sum\ \int$ $\sqrt{\ }\ \overline{\ }$
	Fonctions		cos sin log	cos sin log lim
	Autres	.	$.\ \rightarrow$	$.\ \rightarrow \ldots \infty,$

Table 5.1 – Caractéristiques des trois sous-corpus constituant la base *HAMEX*

Sur la figure 5.2 sont représentés quelques exemples d'EMs de chaque sous-corpus. Une fois la définition du corpus arrêtée, il est nécessaire de présenter les deux protocoles de collecte des données, à savoir : la collecte des tracés manuscrits en-ligne et celle des descriptions par la parole correspondantes à chacune des EMs du corpus. Ceci est l'objet de la section 5.3.

5.3 Collecte des données

La collecte de *HAMEX* s'est faite de telle sorte à avoir autant de scripteurs/locuteurs différents que possible. Chacun d'entre eux a écrit/dicté des EMs des trois sous-corpus présentés dans la section 5.2. Pour ce faire, toutes les expressions ont été mises en commun et un tirage aléatoire a été réalisé pour former des formulaires contenant chacun 75 expressions. Ceci fait que 58 locuteurs et 58 scripteurs ont été requis. Chaque locuteur/scripteur est en charge de dicter/écrire un formulaire complet. Dans les deux sous-sections 5.3.1 et 5.3.2 sont présentés les modes opératoires de la collecte des tracés manuscrits et des signaux audio respectivement.

$48 - 196 < -236$

$47 \pm 155 + (54/120)$

$((15 \times 131)/116)/((75 \times 22) - 169) \neq 0$

a. `Calculette`

$x^2 + 2x + 1 = 0$

$\cos(a + b) = \cos a \cos b - \sin a \sin b$

$\sqrt{3} = \frac{11\alpha - \alpha^3}{2}$

b. `WIKIEM`

$1 - 1 + 1 - 1 + \ldots$

$\forall x, \forall y, f(x) = f(y) \rightarrow x = y$

$\lim_{x \to -\infty} \frac{1}{x^n} = \lim_{x \to +\infty} \frac{1}{x^n} = 0$

c. `WIKIEM-EXT`

FIGURE 5.2 – Quelques exemples d'expressions issues des différents sous-corpus

5.3.1 Collecte des données manuscrites

Pour la collecte de la version manuscrite des EMs de *HAMEX*, nous avons exclusivement utilisé un stylo numérique, le *livescribe digital Smartpen*[1], et du papier numérique du type *anoto*[2] [Liv]. Sur la figure 5.3a est présenté le processus de saisie du tracé manuscrit. La figure 5.3b montre le stylo *livescribe smartpen* embarquant la caméra infrarouge en charge de la capture du motif (pattern en anglais) qui est utilisé par le processeur du stylo pour déduire les informations de trajectoire (positions *(x,y)*, temps *(t)*, pression *(p)*). Sur la figure 5.3c est donné un exemple de feuille *anoto*, un zoom sur le motif de la trame de fond permettant la localisation de l'encre est montré. Enfin, la figure 5.3d, illustre l'exploitation du pattern de la feuille *anoto* par le stylo grâce aux images prises par sa caméra et analysées par son processeur.

1. http ://www.livescribe.com/fr/smartpen/echo/
2. http ://cerig.efpg.inpg.fr/Note/2001/papier_numerique.htm

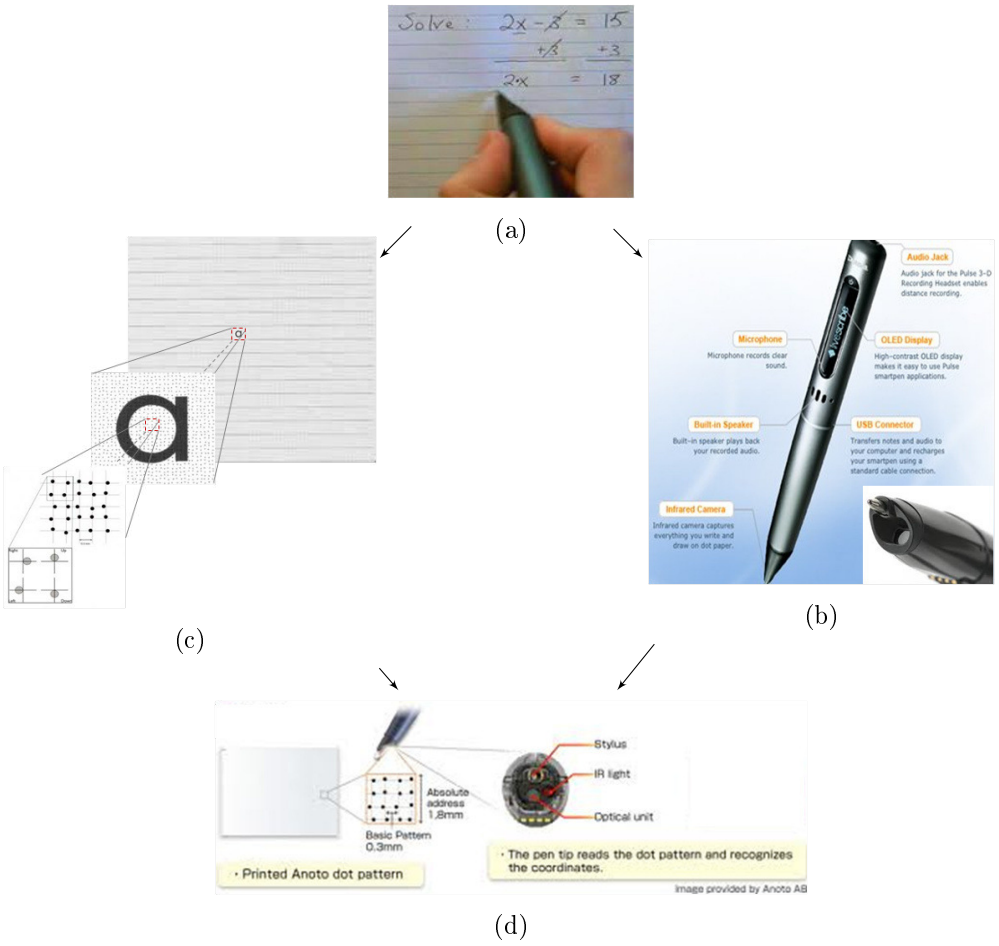


FIGURE 5.3 – Support utilisé lors de la collecte des expressions manuscrites en-ligne. (a) Opération de collecte. (b) Stylo numérique du type *livescribe smartpen*, montrant la caméra infrarouge. (c) Papier numérique type *anoto*, zoom sur le motif. (d) Exploitation du motif par le stylo numérique.

Concernant le processus de collecte lui-même, nous avons adopté une présentation de chacun des formulaires (pour chaque scripteur) sous forme d'un fichier *PDF*. Ce dernier est organisé de façon à présenter sur chaque page cinq expressions. L'une d'entre elle appartient au sous-corpus `Calculette`, deux des quatre restantes sont issues de `WIKIEM` et les deux dernières appartiennent à `WIKIEM-EXT`. La structure de chaque page est faite en alternant un champ où l'EM est présentée au scripteur en format LaTeX et un champ vide où la version manuscrite doit être reportée. On montre sur la figure 5.4 un exemple de la première page. Chaque formulaire est constitué de 15 pages du type de celle décrite ici. De plus, pour chaque formulaire, sur l'entête de la première page, sont ajoutés des champs pour collecter des méta-données concernant le scripteur. Il s'agit d'informations donnant le nom/prénom du scripteur, son âge, sa latéralisation (gaucher/droitier) ainsi que son genre

(figure 5.4).

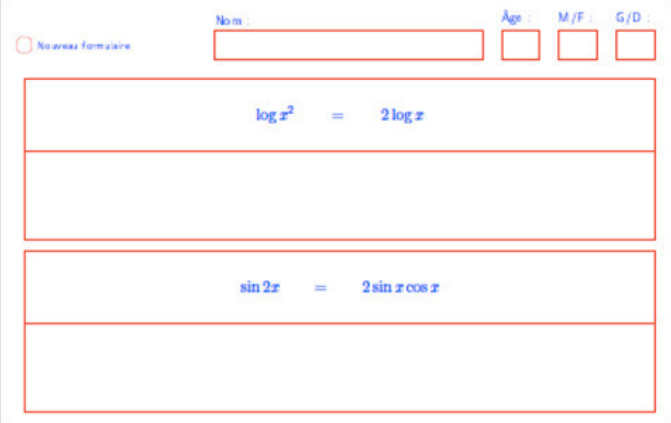


FIGURE 5.4 – Une partie de la structure de la première page du formulaire à remplir en écrit

Par la suite, les scripteurs sont invités à remplir des formulaires composés de feuilles *anoto* sur lesquelles la structure de la figure 5.4 est sur-imprimée. On montre en figure 5.5, un exemple de feuille présentée au scripteur pour la collecte. À l'issue

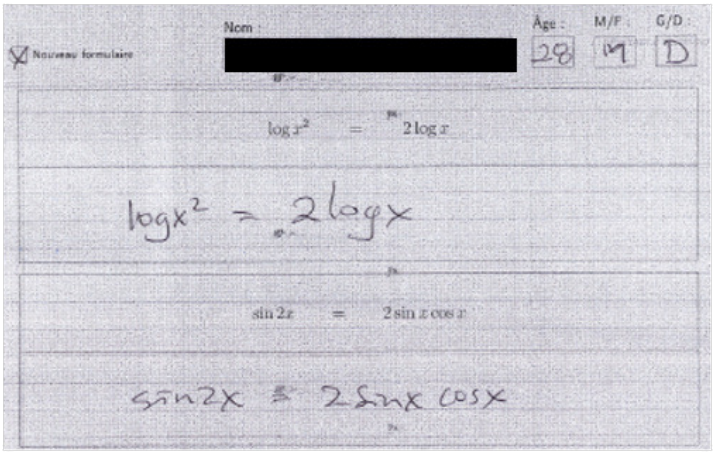
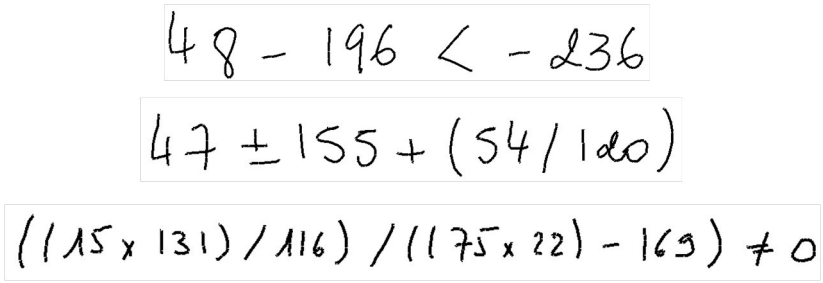
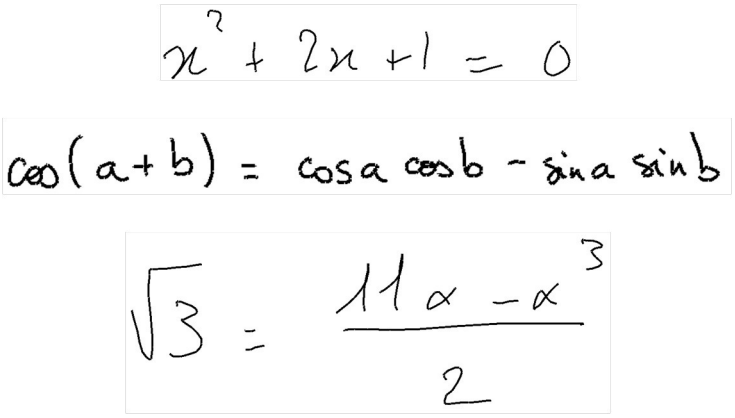


FIGURE 5.5 – Version de feuille *anoto* ayant servi à la collecte, correspondant à la structure présentée en figure 5.4

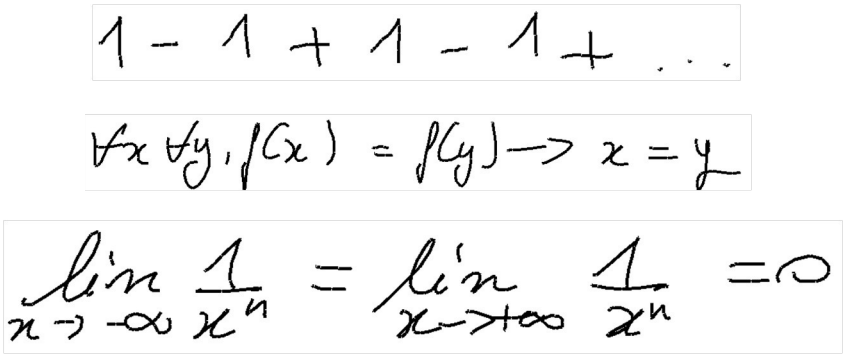
de la collecte, on récupère un fichier du type *unipen* [GSP+94], donnant les données brutes : la séquence de points *(x,y)* regroupés en strokes pour chaque expression. Il est par la suite nécessaire de partir de ces données brutes et de rajouter les annotations nécessaires pour disposer de la vérité terrain. Cette nouvelle présentation des données sera sauvegardée sous un nouveau format, le format *InkML*. Cela fera l'objet de la section 5.4.1. Sur la figure 5.6 sont donnés quelques exemples d'expressions collectées.



(a) Calculette



(b) WIKIEM



(c) WIKIEM-EXT

FIGURE 5.6 – Quelques exemple d'encres collectées correspondantes aux expressions de la figure 5.2

5.3.2 Collecte des données audio

Les 58 formulaires créés précédemment sont conservés pour la collecte des signaux audio. Toutefois, la présentation de l'expression à dicter, à un instant donné, est cette fois-ci faite sur une seule page à la fois. En plus des 75 EMs de chaque formulaire, les 74 symboles du vocabulaire (tableau 5.1) sont également présentés au locuteur de la même manière, afin de disposer d'une base de symboles isolés

en audio. Le formulaire complet est par la suite projeté sur un écran permettant au locuteur de visualiser l'EM à dicter et de passer à la suivante une fois la dictée terminée, en marquant une pause entre deux expressions. L'ensemble des EMs du formulaire, symboles isolés compris, sont enregistrés dans un même signal (un fichier *WAV* classique en mode mono, cadencé à une fréquence de $48kHz$). C'est durant la phase d'annotation que les signaux de chacune des EMs sont découpés (section 5.4.2).

La figure 5.7 présente l'environnement ayant servi à la collecte audio. En plus de l'écran de visualisation de l'EM courante, un microphone conventionnel, le **Lem DO21B** et un enregistreur professionnel classique, le **Marantz PMD 661** ont été utilisés.

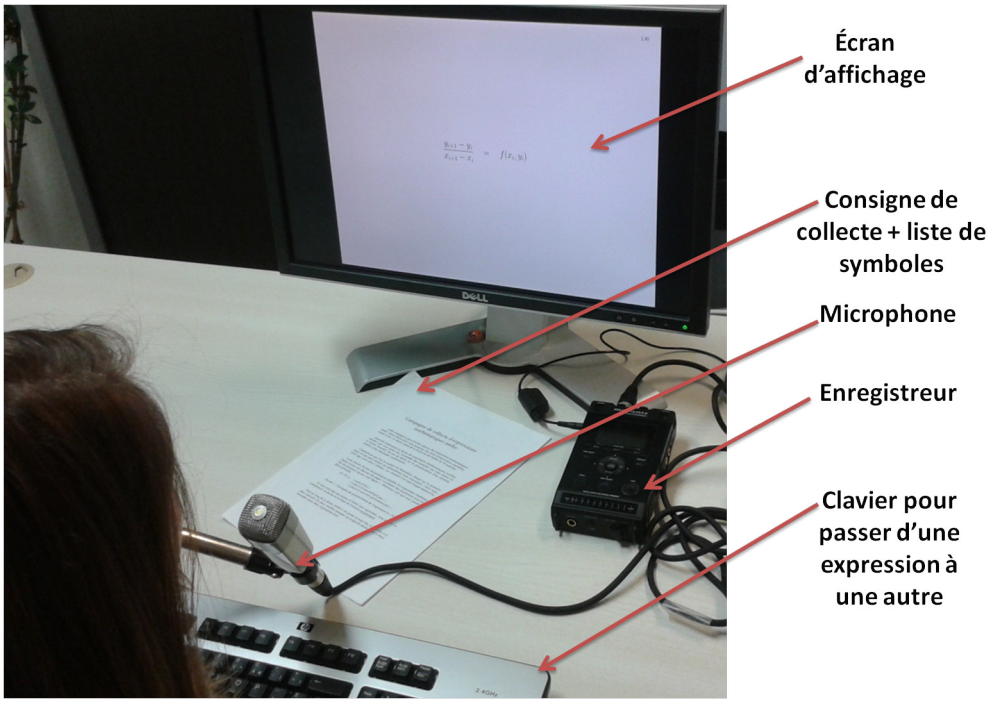


FIGURE 5.7 – Environnement de la collecte audio

Durant le processus de collecte, avant le début de l'opération, la liste de tous les symboles est présentée aux locuteurs afin de s'assurer qu'ils leur soient tous familiers. Cette liste est laissée à leur disposition pendant toute la durée de la collecte et ils peuvent à tout moment la consulter en cas de doute. En outre, quelques exemples d'EMs leur sont proposées pour leur permettre d'accomplir une courte phase d'entraînement. Mis à part cela, les locuteurs sont libres de dicter à leur façon les diverses EMs et aucune règle de dictée ne leur a été imposée. Les deux seules directives suggérées aux locuteurs étaient d'articuler suffisamment bien et de marquer de légères pauses entre les EMs pour faciliter la phase d'annotation (découpage du signal complet en EMs). Dans ces conditions, avec un minimum de contraintes pour le locuteur, on est assuré d'obtenir des dictées les plus naturelles

possibles, avec une certaine variabilité dans les expressions. Par exemple, l'expression $\frac{(a+b)}{2}$ peut être dictée comme suit : 1. *a plus b sur deux*, 2. *a plus b le tout sur deux*, 3. *a plus b entre parenthèses sur deux*, 4. *ouvrir la parenthèse a plus b fermer la parenthèse sur deux*, ...

5.4 Étiquetage des données

Une fois collectées, les 4 350 expressions contenues dans les 58 formulaires (correspondant à $12h$ de signal audio) sont divisées en une base d'apprentissage (formulaire 1 à 39 pour 2 925 EMs) et une base de test (formulaires 40 à 58 donnant 1 425 EMs). De cette façon, les données provenant d'un même scripteur/locuteur seront soit dans la base d'apprentissage, soit dans la base de test. Dans les deux sous-sections qui vont suivre nous allons présenter les protocoles d'annotation des données manuscrites et audio.

5.4.1 Étiquetage des données manuscrites

L'annotation d'une expression manuscrite consiste, à partir du tracé brut disponible au format *Unipen*, à étiqueter chacun de ces traits et à identifier les symboles et les relations qui la composent. Toutes ces informations sont stockées au sein d'un fichier dont le format est adapté à une représentation de type *XML*. Il s'agit du format standard InkML de la norme *W3C*[3]. La structure de ce type de fichier est définie par trois niveaux :

- L'encre en question (une copie des points regroupés en traits issus du fichier *Unipen*). Chaque trait se voit assigné un identifiant unique.
- La vérité terrain au niveau symboles. Il s'agit de regrouper les traits appartenant au même symbole (segmentation) et de donner à chaque symbole l'étiquette associée. De même, chaque symbole se voit assigné un identifiant unique et pointe sur les identifiants des traits qui le composent.
- La vérité terrain au niveau expression. Cela revient à définir les relations liant les différents symboles. Elles sont représentées au sein d'un arbre qui donne la structure MathML[4].

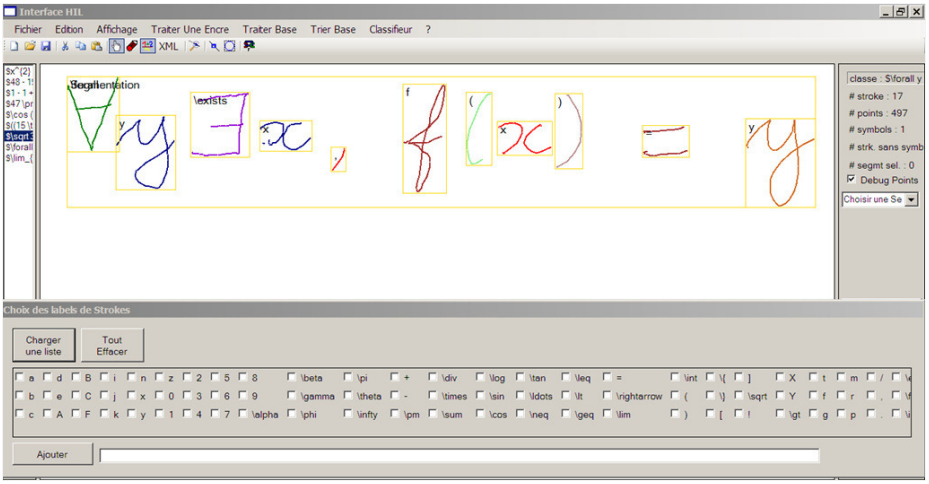
L'entête du fichier InkML regroupe les informations d'identification du scripteur collectées (nom/prénom, age, sexe, latéralisation...). Elle comporte aussi la chaîne LaTeX vérité terrain ainsi que le type d'information contenues dans l'encre (seulement les coordonnées *(x,y)* ou éventuellement les informations de temps, de pression...). On donne sur la figure 5.8 un exemple de fichier InkML représentant l'expression $a < \frac{b}{c}$.

3. http ://www.w3.org/2003/InkML
4. http ://www.w3.org/1998/Math/MathML

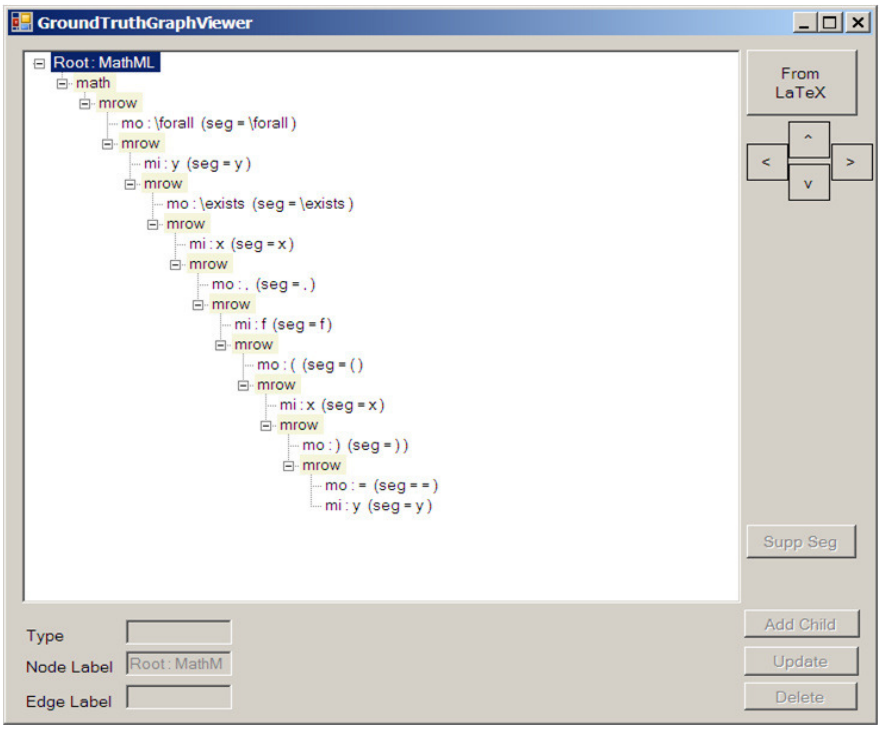
```
<ink xmlns="http://www.w3.org/2003/InkML">
<traceFormat>
<channel name="X" type="decimal"/>
<channel name="Y" type="decimal"/>
</traceFormat>
<annotation type="writer">w123</annotation>
<annotation type="truth">$a&lt;\frac{b}{c}$</annotation>
<annotationXML type="truth" encoding="Content-MathML">
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi xml:id="A">a</mi>
    <mrow>
      <mo xml:id="B">&lt;</mo>
      <mfrac xml:id="C">
        <mi xml:id="D">b</mi>
        <mi xml:id="E">c</mi>
      </mfrac>
    </mrow>
  </mrow>
</math>
</annotationXML>
<trace id="1">985 3317, ..., 1019 3340</trace>
...
<trace id="6">1123 3308, ..., 1127 3365</trace>
<traceGroup xml:id="7">
<annotation type="truth">Ground truth</annotation>
  <traceGroup xml:id="8">
    <annotation type="truth">a</annotation>
    <annotationXML href="A"/>
    <traceView traceDataRef="1"/>
    <traceView traceDataRef="2"/>
  </traceGroup>
  ...
  </traceGroup>
</ink>
```

FIGURE 5.8 – Exemple de fichier de sauvegarde de la vérité terrain de l'écrit (fichier **INKML**)

La création des vérités terrain au niveau symbole et de l'arbre MATHML est une tâche fastidieuse à accomplir à la main. Un outil d'aide à l'annotation a été mis en place dans l'équipe *IVC*. Cet outil rend le processus d'annotation semi-automatique et permet de gagner du temps et de rendre l'annotation plus conviviale. Sur la figure 5.9, une capture de cette interface est présentée.



(a) Création de la segmentation



(b) Création de l'arbre MathML

FIGURE 5.9 – Interface d'annotation des expressions manuscrites en-ligne développé dans l'équipe *IVC*

5.4.2 Étiquetage des données audio

Au niveau du signal audio, l'annotation consiste en deux opérations. La première concerne le découpage du fichier Wav, correspondant à un formulaire complet, en autant de signaux ou segments que d'EMs (en plus des symboles isolés) qui le constituent. Par la suite, il convient de transcrire chacun des segments par le texte tel qu'il a été dicté par le locuteur. C'est cela qui constitue la vérité terrain du côté de l'audio. Celle-ci est donc moins fine que celle produite sur le signal écrit où chaque trait était étiqueté avec son symbole associé. Ici il n'y a pas de segmentation au niveau symbole. Le tableau 5.2 donne quelques exemples de transcriptions de la base *HAMEX*.

L'expression mathématique	Transcription du signal audio correspondante
$47 \pm 155 + (54/120)$	quarante sept plus ou moins cent cinquante cinq plus cinquante quatre divisé par cent vingt
$\sqrt{3} = \frac{11\alpha - \alpha^3}{2}$	racine de trois est égal à onze alpha moins alpha cube le tout sur deux
$\lim_{x \to -\infty} \frac{1}{x^n} = \lim_{x \to +\infty} \frac{1}{x^n} = 0$	limite quand x tend vers moins l'infini de un sur x puissance n égale limite quand x tend vers plus l'infini de un sur x puissance n égal zéro
$\forall x, \forall y, f(x) = f(y) \rightarrow x = y$	pour tout x pour tout y f de x égale f de y tend vers x égale y

TABLE 5.2 – Exemples d'annotations réelles

La tâche de découpage/transcription est accomplie par le biais de l'interface d'aide à l'annotation *Transcriber*[5] dédiée à de telles tâches. Elle permet notamment de signaler tout type de bruit qui serait présent dans le signal. Une capture de l'interface *Transcriber* est donnée en figure 5.10.

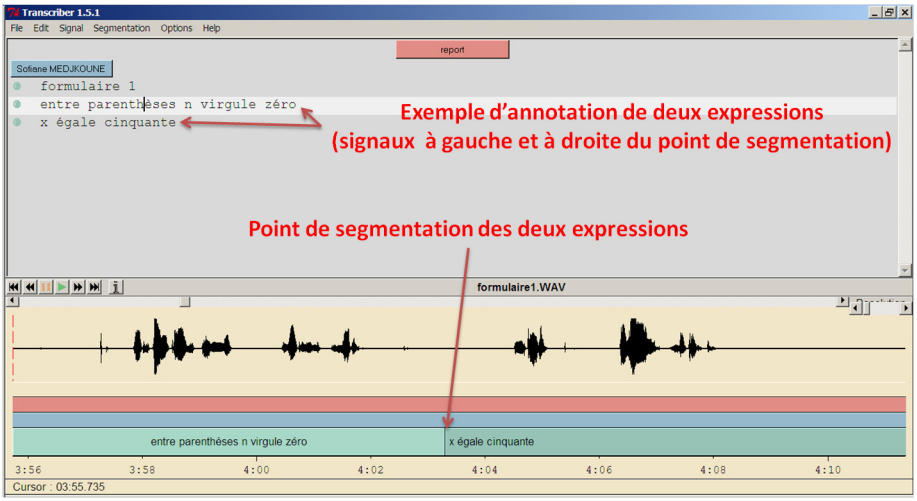


FIGURE 5.10 – Capture d'écran de l'interface de transcription des signaux audio *"Transcriber"*

Une fois les tâches de segmentation et de transcription terminées, l'ensemble des informations (points de segmentation et texte de la vérité terrain) est sauvegardé dans un même fichier au format *trs*. Il est similaire à celui utilisé en écrit. Il est également de type *XML* et permet de faire le lien entre le signal audio et la transcription. Il contient en son entête les méta-données liées au locuteur : nom/prénom, genre,

5. http ://trans.sourceforge.net/en/presentation.php

age ainsi que la langue maternelle, qui sont donnés par le locuteur au début de la dictée du formulaire. Des informations sur les conditions d'enregistrement (studio dans notre cas) ainsi que la qualité de la prononciation (natif ou pas) sont reportées sur l'entête de ces fichiers. Un exemple de la structure du fichier *trs* est donné sur la figure 5.11.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
<Trans scribe="Administrateur" audio_filename="formulaire1" version="1" version_date="120709">
<Speakers>
  <Speaker id="spk1" name="Sofiane MEDJKOUNE" check="no" dialect="native" accent="" scope="local"/>
</Speakers>
<Episode>
  <Section type="report" startTime="0" endTime="1202.72">
    <Turn startTime="0" endTime="1202.72" speaker="spk1" mode="spontaneous" fidelity="high" channel="studio
        ">
      <Sync time="0"/>
        formulaire 1
      <Sync time="235.735"/>
        entre parenthèses n virgule zéro
      <Sync time="243.303"/>
        x égale cinquante
    </Turn>
  </Section>
</Episode>
</Trans>
```

FIGURE 5.11 – Exemple de fichier de sauvegarde de la transcription audio (fichier **trs**)

Une fois collectées et étiquetées, les données (audio et écrites) sont organisées en deux sous-bases. Il s'agit de définir un découpage des données à utiliser en apprentissage et celles à utiliser en test, afin de fixer un cadre permettant de comparer divers systèmes qui utiliseraient *HAMEX* pour valider leurs approches. Dans le tableau 5.3 est donnée l'organisation de ces deux sous-bases.

Base	**# d'expressions**	**# locuteurs/ scripteurs**	**Durée totale de l'enregistrement audio**
Apprentissage	2925	39	$8h$
Test	1425	19	$4h$

TABLE 5.3 – Répartition en apprentissage/test des données collectées (audio et manuscrites)

5.5 Bilan

On a dédié ce chapitre à la présentation de la base *HAMEX*, qui représente la première base de ce genre à notre connaissance. Cette base va servir à toutes les expérimentations et résultats qui seront rapportés dans ce manuscrit de thèse.

Dans les chapitres qui vont suivre nous allons aborder le cœur de ces travaux de thèse en présentant les différentes pistes explorées et en rapportant les résultats obtenus.

Chapitre 6

Étude préliminaire pour la reconnaissance bi-modale des EMs

Sommaire

Ce chapitre est consacré à la pré-étude du problème de reconnaissance des EMs dans le contexte bi-modal. Nous présentons d'abord l'intérêt et les possibilités de fusion entre les signaux écrits et audio. Ensuite, la description de la première expérimentation de validation et les résultats associés sont rapportés.

6.1 Introduction

Nous allons maintenant développer les travaux effectués concernant la fusion des flux manuscrit et audio pour la reconnaissance des expressions mathématiques.

Dans un premier temps, nous commençons par présenter le problème de fusion dans le cadre de la reconnaissance des EMs, en soulignant la complémentarité existante entre ces deux modalités. Ensuite nous identifierons les possibilités de fusion qui s'offrent à nous dans le cadre du problème traité ici.

Dans un second temps, nous proposerons les premières expérimentations menées pour quantifier l'apport de la fusion de données incluant les signaux écrit et sonore. Nous traiterons le cas de la reconnaissance bi-modale de symboles isolés en le considérant comme un sous-problème de la reconnaissance des EMs complètes. Les problèmes de segmentation et d'interprétation sont donc ici ignorés, on fournit en entrée du système un signal correspondant à la totalité d'un symbole manuscrit et son équivalent prononcé par un locuteur. Pour la base de symboles isolés en écrit, nous avons eu recours à la base CIEL [1] de symboles isolés [Awa10], disponible au laboratoire. En revanche, pour le cas des symboles isolés en audio, lors de la collecte de la base *HAMEX*, comme nous l'avons déjà précisé, les locuteurs sont invités à dicter l'ensemble des 74 symboles du vocabulaire composant cette base.

6.2 Considérations pratiques et situation du problème

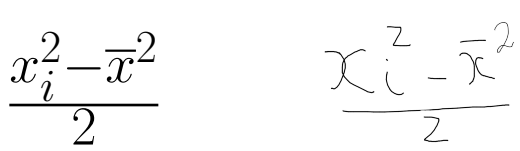
Avant d'aborder la mise en place du système de reconnaissance bi-modale des symboles mathématiques isolés, nous allons présenter quelques considérations sur lesquelles va s'appuyer le reste de ce manuscrit. Il s'agit de choix adoptés tout au long des réalisations de cette thèse.

6.2.1 Complémentarités des flux écrit et audio

Dès les chapitres 1, 2, et 3, à travers les exemples que nous avons apportés, il ressort qu'une forte complémentarité existe entre les flux audio et manuscrit en-ligne. Cette complémentarité a notamment été démontrée par plusieurs travaux et pour différentes problématiques [WVDM96, Kai05, Kal05, HHI07, Kor08]. Dans le cas des EMs, cette complémentarité parole-écriture manuscrite est notamment remarquable dans le cas où une des modalités rencontre des difficultés et que, grâce à la présence de la seconde modalité, l'ambiguïté est levée. Pour illustrer ce propos, sur la figure 6.1 est présenté un exemple de complémentarité des deux modalités.

Dans cet exemple, si on considère uniquement le signal manuscrit en-ligne (figure 6.1b), plusieurs problèmes peuvent rendre difficile l'interprétation. Par exemple, deux des symboles "2" au numérateur et au dénominateur peuvent être très facilement confondus avec le symbole "z". Si on consulte la transcription automatique de

1. Cette base est composée de symboles isolés manuscrits en-ligne. Elle a été collectée au sein du laboratoire *IRCCyN* (IVC) dans le cadre du projet ANR *CIEL (Conversion Indexation de l'Écriture en Ligne)* avec le partenaire industriel *VisionObjects* (http ://www.visionobjects.com)



(a) Vérité terrain de l'EM (rendu LaTeX)

(b) Tracé manuscrit d'une EM

"x indice i au carré moins x barre au carré sur deux"

(c) Transcription automatique de la dictée de l'EM

FIGURE 6.1 – Exemple de complémentarité entre les flux audio et manuscrit en-ligne

la figure 6.1c, dans le premier cas, c'est le mot *carré* qui permet de lever l'ambiguïté de l'identité de ce symbole. L'identité du second, quant à elle, est rendu moins confuse grâce au mot *deux*. Au niveau relation, la position du "i" relativement au "x" est très ambigüe en écrit. En audio celle-ci est explicitement désignée comme un *indice*, ce qui permet d'identifier clairement la nature de cette relation. Au niveau de la transcription automatique de la dictée de l'EM, si tous les symboles composant l'EM sont présents, leur disposition dans un espace $2D$ est loin d'être évidente (par exemple, on pourrait croire que c'est i qui est au carré et non le symbole x). En consultant la modalité manuscrite en-ligne, la situation devient moins confuse.

La figure 6.2 donne différents exemples d'interprétations possibles pour chacune des deux modalités. Elle montre également comment le traitement bi-modal peut assurer une bonne interprétation finale, quand les traitements mono-modaux s'avèrent infructueux.

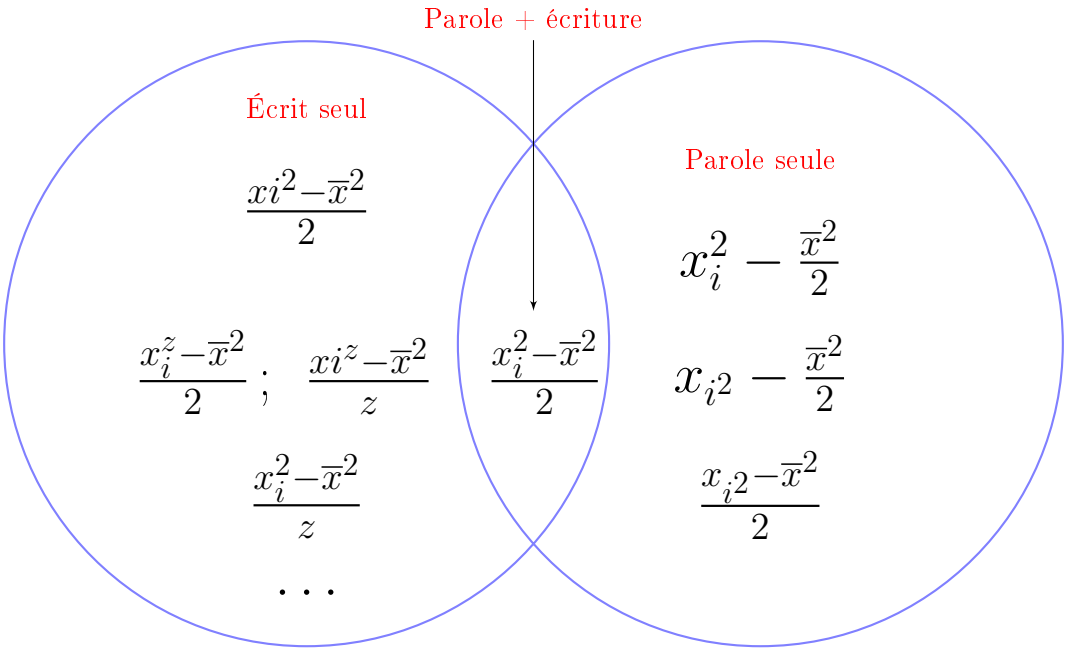


FIGURE 6.2 – Espace des solutions possibles dans chacune des deux modalités : la meilleure solution est celle qui est commune. Les solutions erronées le sont pour des motifs différents suivant la modalité

6.2.2 Niveaux et méthodes de combinaison envisageables

Nous avons vu dans la section 4.3 de façon générale qu'il était possible de combiner les informations à plusieurs niveaux (fusion précoce, tardive ou hybride). Dans le cas particulier des EMs, la fusion peut également intervenir sur deux niveaux spécifiques : **symbole** ou **relation**. Au niveau symbole, il s'agit dans ce cas de présenter au module en charge de l'interprétation de l'EM les meilleures hypothèses de symbole issues d'un raisonnement bi-modal. Aussi bien pour le système écrit que pour le système audio, agir à ce niveau assure un meilleur découpage du signal brut (segmentation) et un meilleur étiquetage des hypothèses segmentées (reconnaissance plus fiable). De même, au niveau des relations lorsque l'on cherche à interpréter l'EM, la connaissance des deux modalités va permettre de lever des ambigüités. Une fusion à ce niveau permet de faire le tri parmi toutes les interprétations possibles et de choisir parmi celles qui sont issues de la mise en commun des solutions proposées au niveau de chacune des deux modalités.

A ce stade, une fois les deux niveaux précédents (symbole et relation) identifiés, nous devons sélectionner, parmi les niveaux de fusion (précoce, tardive, hybride) et les méthodes évoqués dans le chapitre 4 les plus adaptés a priori à notre problème de fusion d'EMs.

La nature hétérogène des signaux des deux modalités et la difficulté de leur synchronisation nous apportent un élément de réponse concernant le niveau de fusion. En effet, dans ce cas de figure il est assez naturel de procéder à une fusion de décisions, donc tardive. Ceci permet de s'affranchir des problèmes d'hétérogénéité et de synchronisation des deux flux.

Le choix du niveau nous impose des contraintes pour le choix des méthodes possibles. En effet, puisque nous envisageons une fusion tardive, une fusion à base d'estimateurs est à exclure (tel que le témoignent les travaux rapportés dans la littérature et dans la mesure où le problème que nous avons à traiter se situe dans un contexte de classification et non d'estimation). De ce fait, des méthodes de fusion à base de règles et à base de méthodes de classification seront préférées dans les travaux rapportés dans cette thèse.

6.2.3 Positionnement de la solution proposée

La solution apportée dans cette thèse est construite autours de deux constatations issues de l'étude bibliographique.

La première concerne la différence de maturité des systèmes de reconnaissance automatique des EMs proposés pour chacune des deux modalités. En effet, comme nous l'avons vu dans la première partie de ce manuscrit, la communauté de la reconnaissance automatique de l'écriture manuscrite a proposé des solutions, de loin plus abouties et plus diversifiées que ce que propose la communauté de transcription automatique de la parole.

La seconde constatation réside dans le fait que le caractère $2D$ des EMs est naturellement présent dans le cas du tracé manuscrit en-ligne, quand celui-ci est

caché dans le cas de la description par la parole.

Au vu de ces deux points, nous proposons, dans ce travail, de considérer une modalité principale qui constitue le cœur de notre système : l'écriture manuscrite en-ligne. C'est autour de cette modalité que toute la chaîne du processus de reconnaissance des expressions mathématiques va être structurée. La modalité de la parole va, dans le cadre de notre approche, jouer le rôle d'aide à la reconnaissance. Les informations issues de la parole vont être exploitées dans une optique de désambiguïsation du processus de reconnaissance à chaque fois que cela est nécessaire.

Le chapitre suivant reviendra sur cette architecture pour résoudre l'interprétation d'EMs complètes. Auparavant, nous étudions dans la section 6.3 le problème simplifié de la reconnaissance des symboles isolés.

6.3 Reconnaissance bi-modale des symboles mathématiques isolés

Afin de mettre en place ce système bi-modal de classification de symboles isolés, et en considérant une fusion tardive (*cf.* section 6.2.2), il est important de disposer de deux systèmes experts. Chacun de ces deux systèmes est en charge de traiter une des deux modalités et de fournir les décisions intermédiaires associées. Nous présentons en section 6.3.1 le système utilisé en écrit ainsi que ces performances. La section 6.3.2 est quant à elle dédiée à la description et à la présentation des performances du système audio. La section 6.3.3 rapporte les différentes propositions de combinaison des deux flux et également les résultats associés.

6.3.1 Reconnaissance des symboles isolés manuscrits en-ligne

Nous commençons d'abord par présenter le moteur de reconnaissance sur lequel nous nous sommes appuyés pour accomplir cette tâche. Par la suite, dans une seconde sous-section, nous allons discuter de ces performances de reconnaissance.

6.3.1.1 Système de reconnaissance utilisé pour la reconnaissance du signal manuscrit

Dans la section 2.4 sont rapportées différentes techniques de reconnaissance de symboles. Dans notre cas nous allons faire appel à celle mise en œuvre dans le système de reconnaissance des EMs développé dans les travaux de thèse d'Awal [AMVG12].

En effet, nous souhaitons disposer d'un classifieur qui puisse produire un score de reconnaissance associé aux classes d'une liste des $N - meilleures$ solutions. Agir de la sorte peut aider à lever des ambiguïtés d'étiquetage, soit grâce au processus de fusion (comme dans ce chapitre), soit grâce à l'étape d'interprétation dans le cas de la reconnaissance mono-modale [RSTS06, RK09, AMVG12].

Pour satisfaire la première condition, des classifieurs avec des sorties probabilistes sont utilisés (il est également possible de considérer des classifieurs flous ou basés sur des fonctions de croyance, qui ne sont pas l'objet de notre étude), en ayant

recours à un étage de sortie approprié assurant une normalisation des sorties (que l'architecture du classifieur soit un *PMC* ou un *TDNN*). Cet étage assure que pour un symbole donné hs_i, le score qu'il se voit assigné l'étiquette C_j est donné par la probabilité $P(C_j|hs_i)$, avec la contrainte supplémentaire $\sum_{j=1}^{Nb_{Classes}} P(C_j|hs_i) = 1$.

La réponse quant à la seconde condition est de prendre une liste des $topN$ meilleurs candidats. La valeur de $topN$ est choisie de telle sorte à satisfaire l'inéquation 6.1 :

$$\begin{cases} \sum_{j=1}^{topN} P(C_j|hs_i) \leq ScoreCum_{max}, \\ \\ \qquad topN \leq topN_{max}. \end{cases} \tag{6.1}$$

Dans l'équation 6.1, $ScoreCum_{max}$ est un seuil sur les scores cumulés qui permet de ne garder que les candidats les plus probables et ne pas prendre systématiquement $topN_{max}$ candidats ; $topN_{max}$ a pour but de limiter le nombre maximal de candidats à une borne supérieure si la somme des scores n'atteint pas $ScoreCum_{max}$. Ces deux paramètres sont fixés de façon expérimentale dans notre cas.

Les classifieurs qui ont le meilleur comportement dans le système de Awal et al [AMVG12], en charge du traitement de la modalité écrite dans notre système, sont de type réseau de neurones (*PMC* ou *TDNN*). De ce fait, dans le cas de la reconnaissance des symboles isolés, ce sont ces deux classifieurs que nous avons mis en place pour traiter les signaux manuscrits en-ligne représentant chaque symbole. Le succès d'un tel système réside principalement en trois points clés :

1. **Espace de représentation des données suffisamment discriminant :** En effet, il est important de disposer d'entrées représentées dans un espace offrant une variance inter-classes la plus importante possible. Ceci a pour effet de réduire la complexité des frontières séparant les classes et donc par la même la complexité du modèle à trouver pour effectuer la tâche de classification.
2. **Choix d'une structure adéquate du réseau :** Ce point est également très important. En effet, à l'image du *RN* de la figure A.1a, plusieurs paramètres sont à ajuster de façon à proposer la structure minimale (en taille) et suffisante (en performances) pouvant répondre au problème traité. Ces paramètres définissent la *topologie* du réseau et sont le nombre de couches et le nombre de neurones de chacune des couches. Ceci détermine le nombre de poids (paramètres libres) à fixer de façon optimale. Le choix du type des fonctions d'activation peut également être déterminant.
3. **Base d'exemples d'apprentissage suffisamment complète :** Une fois la topologie du réseau choisie, il convient donc de faire l'apprentissage du *RN* pour résumer la connaissance contenue dans la base d'apprentissage dans le modèle final donné par le couple (topologie du classifieur, poids ajustés). Plus la base est représentative, plus le pouvoir de généralisation du classifieur est grand. C'est donc pour cela qu'il est important de disposer de plus de données possible et pour toutes les classes du problème traité. Mais aussi avoir une variabilité au sein d'une même classe qui représenterait la dispersion réelle

(pratique).

Dans la suite nous allons développer les trois points précédemment cités.

Espace de représentation des données suffisamment discriminant

Avant de parler d'espace de représentation des données à reconnaître : les symboles manuscrits en-ligne, il est important de revenir sur les caractéristiques de ce signal. Comme nous l'avons introduit dans la section 2.2, le symbole est formé par un ou plusieurs traits élémentaires, qui eux même sont formés d'un ensemble de points délimités par le poser et le lever de stylo. De ce fait, l'échelle (différence de taille de l'écriture due au scripteur ou au matériel de saisie) ainsi que le nombre de points (différence due à la vitesse d'écriture ou à cause du matériel de capture du signal) pour un symbole donné peuvent être très variables. Ceci complique encore plus la tâche du classifieur en charge de leur étiquetage, en augmentant davantage la dispersion au sein de la même classe. Afin de réduire le biais dû à ces propriétés et avant même d'aller chercher un espace de représentation plus favorable à l'opération de classification, le signal subit un pré-traitement. Le but de ce dernier est de représenter tous les symboles avec le même nombre de points et dans une échelle unique donnée.

Une fois ces pré-traitements accomplis, il est possible par la suite de changer d'espace de représentation des données en passant dans l'espace des caractéristiques. Ce dernier va être plus favorable à la tâche d'annotation automatique des hypothèses de symboles.

Dans la suite, nous allons détailler chacune des étapes ci-dessus.

1. **Échantillonnage :** L'idée est de passer d'une représentation du tracé brute (un ou plusieurs trait(s)) en (x, y) ou chacun des points est acquis en fonction de la fréquence temporelle du dispositif de saisi, à une représentation dans laquelle le tracé est spatialement échantillonné de façon régulière pour contenir exactement Nb_{pts}. Dans cette nouvelle trajectoire, les points sont identifiés par leurs coordonnées (x, y) toujours mais aussi par une troisième dimension qui définit l'état du stylo au moment de son passage par cette position, à savoir : levé ($PenUp$), ou posé ($PenDown$). Cette dimension prend dans notre cas deux valeurs particulières : -1 si le stylo est levé et 1 si le stylo est posé. L'objectif de ce ré-échantillonnage est, comme indiqué plus haut, de produire des exemples de la même classe ayant des représentations les plus proches possibles. Cela assure également la compatibilité des données présentées aux entrées du classifieur. En effet, la couche d'entrée des classifieurs que nous avons adoptés possède un nombre de neurones fixe dans la couche d'entrée en charge de la capture des données (voir figure A.1a). Dans les travaux rapportés dans ce manuscrit, le pas d'échantillonnage est obtenu en fixant Nb_{pts} à 50 (validé expérimentalement) et une interpolation linéaire est utilisée pour déduire les nouvelles coordonnées. Sur l'exemple de la figure 6.3 est montré l'effet du ré-échantillonnage sur deux exemples de tracé du symbole "b".

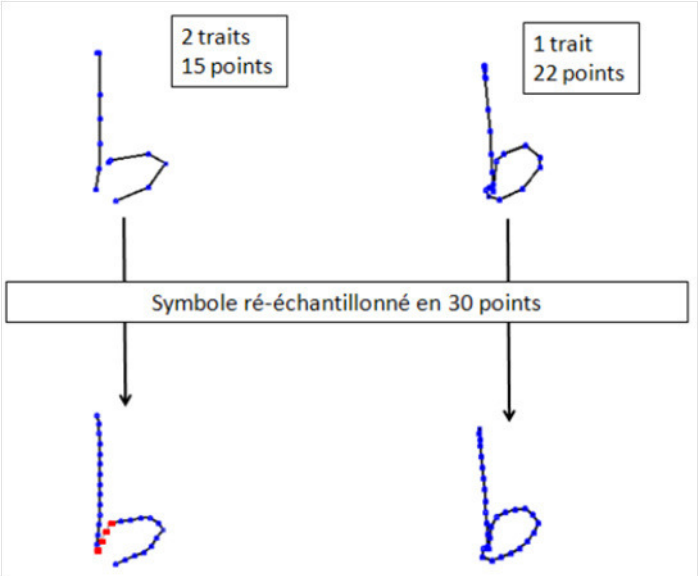


FIGURE 6.3 – Deux exemples de tracé du symbole "*b*" : celui de gauche est écrit en deux traits et avec 15 points au total, celui de droite, écrit en un seul trait, possède 22 points. Après ré-échantillonnage, les deux tracés sont représentés ici par 30 points chacun, régulièrement espacés. Deux types de points sont à distinguer, les ronds en bleu, sont les points ou le stylo est posé. Les carrés en rouge, représentent les points ou le stylo est levé (cas de deux traits) [Poi05, Awa10]

2. **Mise à l'échelle :** Dans un second temps, une fois l'échantillonnage spatial effectué, tous les symboles subissent une mise à l'échelle et un recentrage. Celle-ci va permettre de disposer d'une représentation qui soit à la fois insensible à la translation (grâce au recentrage) et à la taille des symboles (par le biais de la mise à l'échelle). Pour ce faire, le tracé est ramené de façon proportionnelle (garder le rapport de sa hauteur à sa largeur) dans un repère de sorte qu'il soit centré relativement à son plus grand coté dans un carré de dimensions $[-1, 1]^2$.

3. **Extraction des caractéristiques :** Pour chacune des hypothèses de symboles ayant subi les deux traitements précédents, un vecteur de 7 caractéristiques est extrait au niveau de chacun de ces 50 points. Ces caractéristiques sont composées des coordonnées (x, y) elles-mêmes, complétées par les cosinus directeurs pour exprimer les directions en ce point $(\cos\theta, \sin\theta)$ et par l'information de courbure en ce point $(\cos\varphi, \sin\varphi)$. À ces six caractéristiques se rajoute l'information de poser ou de lever de stylo, pour former les sept caractéristiques considérées au niveau de chaque point.

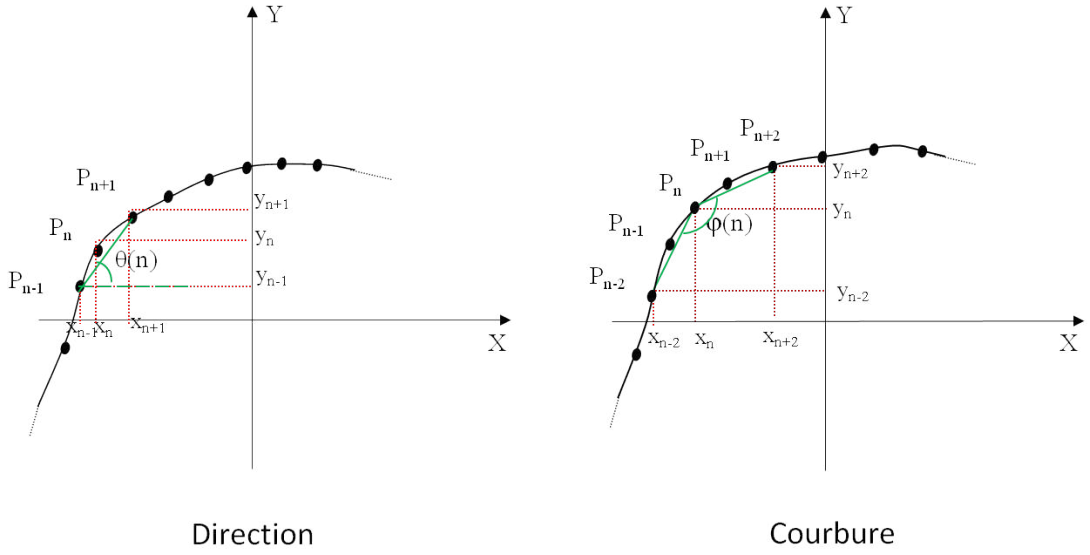


FIGURE 6.4 – Illustration du processus d'extraction des informations de direction $(\cos\theta, \sin\theta)$ et de courbure $(\cos\varphi, \sin\varphi)$ au niveau d'un point P_n [Poi05, Awa10]

À partir de la figure 6.4, les informations de direction et de courbure peuvent être exprimées par les équations 6.2 et 6.3.

$$\begin{cases} direction\ selon\ x = \cos\theta_n = \begin{cases} 0\ si\ x_{n-1} = x_{n+1}, \\ \dfrac{|x_{n-1}, x_{n+1}|}{\| P_{n-1}, P_{n+1} \|}\ sinon. \end{cases} \\ direction\ selon\ y = \sin\theta_n = \begin{cases} 0\ si\ y_{n-1} = y_{n+1}, \\ \dfrac{|y_{n-1}, y_{n+1}|}{\| P_{n-1}, P_{n+1} \|}\ sinon. \end{cases} \end{cases} \tag{6.2}$$

$$\begin{cases} courbure\ selon\ x = \cos\varphi_n = \cos\theta_{n-1} \times \cos\theta_{n+1} + \sin\theta_{n-1} \times \sin\theta_{n+1} \\ courbure\ selon\ y = \sin\varphi_n = \sin\theta_{n-1} \times \cos\theta_{n+1} - \cos\theta_{n-1} \times \sin\theta_{n+1} \end{cases} \tag{6.3}$$

Après caractérisation du tracé correspondant à un symbole, un vecteur de caractéristiques de taille $50 \times 7 = 350$ est obtenu. Ce dernier est fourni en entrée du classifieur et c'est sur ce vecteur qu'est accomplie la reconnaissance.

À présent nous allons décrire les structures des classifieurs retenues pour notre système.

Choix d'une structure adéquate du réseau

Aussi bien pour le *PMC* ou le *TDNN*, la couche d'entrée contiendra 350 neurones (taille du vecteur de caractéristiques). Le nombre de neurones de la couche de sortie, quant à elle, correspond au nombre de classes considérées dans le problème. Dans le cas des symboles isolés, tout le vocabulaire de la base *HAMEX* est considéré, à savoir 74 classes. Le caractère probabiliste des sorties est assuré par la fonction de transfert *softmax* [Tou90] défini comme suit :

$$P(C_j/X) = softmax(\sigma_i(X)) = \frac{e^{\sigma_i(X)}}{\sum_{k=1}^{Nb_{classes}} e^{\sigma_k(X)}}, \tag{6.4}$$

tel que :

$P(C_i/X)$: est la probabilité qu'un vecteur de caractéristiques X soit de la classe C_i

$\sigma_i(X)$: est le potentiel synaptique du $i^{ième}$ neurone de sortie associé à l'exemple X

Il reste à présent à définir la couche, voire les couches, cachée(s). En effet, savoir combien de couches cachées va contenir le *RN* et combien de neurones par couche est crucial. Ceci définit sa complexité et la difficulté de son apprentissage, mais aussi son pouvoir séparateur des différentes classes du problème. Globalement, trois approches existent pour fixer cela (*cf.* annexe A) : une approche empirique, une deuxième approche automatique et une dernière approche orientée méta-apprentissage.

Dans notre cas, c'est la première catégorie de méthodes que nous avons considérée. En effet, nous sommes partis de la configuration optimale proposée par Awal et al [AMVG12], pour la reconnaissance de symboles mathématiques manuscrits. Dans le cas du *PMC*, une seule couche a été considérée, nous avons maintenu cette structure en optimisant le nombre de neurones de la couche (d'autres structures avec plus de couches cachées ont été entrainées mais ne se sont pas avérées être meilleures). Pour ce qui est du *TDNN*, nous avons repris l'architecture optimale proposée initialement dans [Poi05] et reprise par Awal et al [Awa10]. Elle est représentée en figure 6.5. Nous avons également adapté le nombre de neurones de la couche cachée du *PMC* contenue dans ce *TDNN*. En effet, le *TDNN* utilisé ici se compose de deux parties inter-connectées : une partie pour l'*extraction* (typique du *TDNN*) et une seconde pour la *reconnaissance* de type *PMC*. Durant la phase d'extraction, le vecteur de caractéristiques en entrée (de la première couche du *TDNN*) est transformé en un autre vecteur de caractéristiques d'ordre supérieur. En effet, le champ de vision restreint des neurones des couches supérieures permet la détection de caractéristiques topologiques locales. En plus de cela, le partage de poids (suivant la direction temporelle) donne un pouvoir supplémentaire au réseau, qui est celui de la détection de la présence ou non de la même caractéristique suivant la trajectoire du stylo. À un instant donné, nb_feat neurones (*cf.* figure 6.5) sont en charge de visualiser les mêmes neurones de la couche précédentes, procurant ainsi différents points de vue pour un même morceau du signal temporel. Sur la couche de sortie de l'extracteur, de nouvelles caractéristiques sont ainsi obtenues. Ce nouveau vecteur

de caractéristiques est fourni à la partie classifieur (dernière couche de l'extracteur qui est elle même la couche d'entrée du classifieur) pour accomplir la reconnaissance classique (simple *PMC*).

Dans les deux cas, *PMC* ou *TDNN*, les neurones des couches cachées ont pour fonction d'activation la fonction de transfert du type *sigmoïde*, à valeur dans $[-1, 1]$ et infiniment dérivable :

$$Sigmoid(\sigma_i) = \frac{e^{2\sigma_i} - 1}{e^{2\sigma_i} + 1}, \tag{6.5}$$

σ_i *étant le potentiel synaptique au neurone* i.

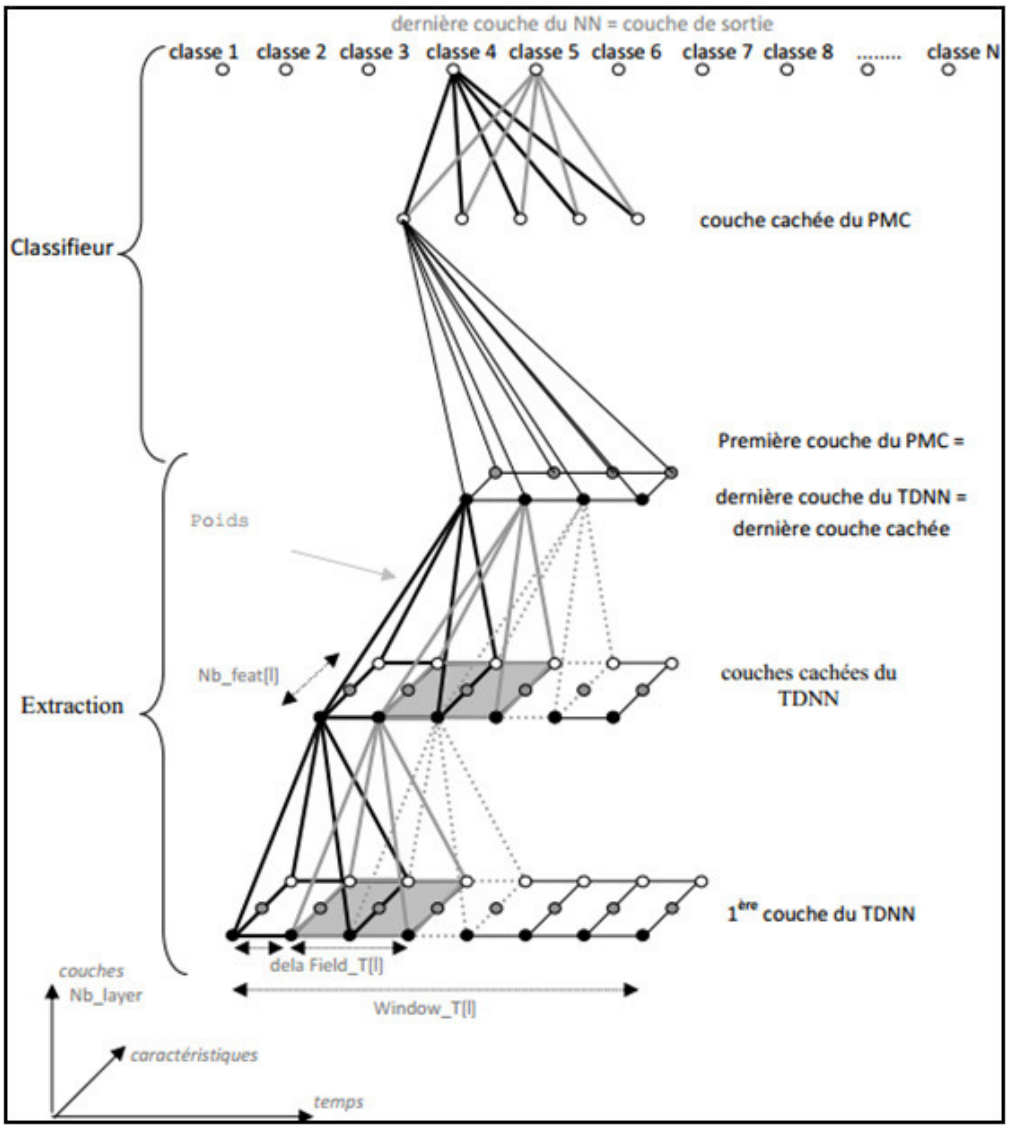


FIGURE 6.5 – Architecture du *TDNN* [Poi05, Awa10]

Une fois les structures des réseaux fixées, il convient de faire leur apprentissage. En fait, il s'agit d'ajuster leurs paramètres libres, les *poids*. En général, ceci se fait par un *apprentissage supervisé*, ce qui est notre cas. Pour cela, l'algorithme classique de rétro-propagation du gradient de la fonction de coût est mis en œuvre (*cf.* annexe A).

Base d'exemples d'apprentissage suffisamment complète

Une fois la structure optimale du classifieur établie, un autre facteur important est la qualité des données utilisées. Ces dernières doivent couvrir tout le vocabulaire du problème traité (toutes les classes de symboles dans le cas de la reconnaissance des symboles isolés). Chaque classe doit avoir le plus de représentants possible et qui soient le plus diversifié possible. Cette dernière contrainte va permettre de couvrir la dispersion au sein de chacune des classes. Cela aura pour effet d'apprendre un modèle pour lequel les frontières sont mieux définies.

Dans notre cas, pour satisfaire à ces pré-requis, nous avons utilisé différentes bases de données. Pour le cas des EMs complètes, nous avons utilisé les données fournies dans le cadre de la compétition *CROHME* pour l'apprentissage du système écrit seul, ainsi que la base bi-modale *HAMEX* présentée dans le chapitre 5 pour tous les aspects de fusion, nous reviendrons sur cela à l'occasion du chapitre suivant sur la reconnaissance des EMs complètes. Concernant les symboles isolés, l'objet du présent chapitre, nous avons fait appel à la base *CIEL* de symboles isolés manuscrits [Awa10]. Cette base est assez conséquente et riche, dans la mesure où chacun de ses symboles est disponible en 280 exemplaires, chacun est écrit par un scripteur différent. Ceci répond à la fois à la question de variabilité au sein d'une même classe et à la question du nombre de représentants par classe (plusieurs scripteurs différents par classe).

À présent, nous allons passer à la présentation détaillée des systèmes utilisés pour traiter le signal manuscrit en-ligne (les deux classifieurs *PMC* et *TDNN*). Nous allons également présenter de façon quantitative la répartition des données utilisées pour l'apprentissage et le test des dits systèmes. Et nous finirons par présenter les résultats obtenus et leurs analyses.

6.3.1.2 Résultats expérimentaux du système écrit seul

Ainsi que nous l'avons déjà indiqué, cette première expérimentation porte sur l'ensemble du vocabulaire de la base *HAMEX*, à savoir les 74 classes. Ainsi, les RNs utilisés seront donc définis avec 74 neurones sur la couche de sortie. Dans la section précédente, nous avons également présenté les caractéristiques que nous avons utilisé tout au long de ce travail, au nombre de 7 par point. Le pré-traitement dont nous avons également parlés précédemment, assure une représentation identique de tous les symboles à reconnaître en ré-échantillonnant chaque forme pour la représenter par 50 points quelque soit sa taille ou le nombre de traits élémentaires qu'elle contient. Ceci fait que le vecteur de caractéristiques pour une hypothèse donnée est de taille $50 \times 7 = 350$. Par conséquent, la couche d'entrée des *RNs* contient 350 neurones.

Sur le tableau 6.1 est donnée le détail de la topologie du classifieur *PMC* utilisé.

# de paramètres libres		$100 \times (350 + 1) + 74 \times (100 + 1) =$ **42 754**	
# couches	# neurones de la couche d'entrée	# neurones de la couche cachée	# neurones de la couche de sortie
3 (1 entrée, 1 cachée, 1 sortie)	350	100	74

TABLE 6.1 – Détails de la structure du réseau de neurones *PMC* utilisé pour la classification des symboles isolés manuscrits en-ligne

Le classifieur de type *TDNN* est quant à lui composé d'**une seule couche cachée** dans la partie extraction (*TDNN*) et d'**une seule couche cachée** pour la partie classifieur (*PMC*). Le détail de sa topologie est rapporté dans la table 6.2, en se référant aux notations adoptées dans la figure 6.5.

On constate qu'à cause de, où grâce à, la contrainte des poids partagés, l'architecture du *TDNN* dispose de moins de paramètres libres que le *PMC*.

<table>
<tr><td colspan="2"># de paramètres libres</td><td colspan="2">$7 \times (20 \times 7 + 1) + 100 \times (7 \times 20 + 1) + 74 \times (100 + 1) =$ 22 561</td></tr>
<tr><td colspan="4">Partie extraction (voir la partie Extraction de la figure 6.5)</td></tr>
<tr><td colspan="4">Couche d'entrée du TDNN (l=0)</td></tr>
<tr><td># neurones selon le temps ($Window\_T[0]$)</td><td># neurones selon les caractéristiques ($nb\_feat[0]$)</td><td># neurones de la fenêtre temporelle (de convolution) $field\_T[0]$</td><td># neurones de délai entre fenêtres $delay[0]$</td></tr>
<tr><td>50</td><td>7</td><td>20</td><td>5</td></tr>
<tr><td colspan="4">Couche cachée du TDNN (l=1) (qui est aussi la couche d'entrée du PMC)</td></tr>
<tr><td colspan="2"># neurones dans la direction temporelle $Window\_T[1]$</td><td colspan="2"># neurones dans la direction caractéristiques $nb\_feat[1]$</td></tr>
<tr><td colspan="2">7</td><td colspan="2">20</td></tr>
<tr><td colspan="4">Partie classifieur (voir la partie Classifieur de la figure 6.5)</td></tr>
<tr><td colspan="2"># neurones de la couche cachée du PMC</td><td colspan="2"># neurones de la couche de sortie du PMC (classes du problème)</td></tr>
<tr><td colspan="2">100</td><td colspan="2">74</td></tr>
</table>

TABLE 6.2 – Détails de la structure du réseau de neurone à convolution *TDNN* utilisé pour la classification des symboles isolés manuscrits en-ligne

Les données manuscrites utilisées dans cette expérimentation sont donc un sous-ensemble des données de la base *CIEL* qui correspondent au vocabulaire de la base *HAMEX* (*cf.* tableau 5.1). Au sein de chaque classe, c'est un double tirage aléatoire qui est effectué parmi les 280 scripteurs [2] la composant. Le premier tirage va répartir les données en données d'apprentissage (deux tiers) et le tiers restant pour le test. Le second tirage, concerne les données d'apprentissage, où deux tiers d'entre elles sont utilisées effectivement pour l'apprentissage et le reste est utilisé pour la validation (pour la validation de la topologie du classifieur et des poids associés). La répartition des données, à l'issue de ces tirages, est rapportée dans le tableau 6.3.

2. Certains exemples de certaines classes sont absents dans la base *CIEL* originale. Ils ont été écartés à l'issue de la vérification des données après collecte(nettoyage de la base) [Awa10].

Base		# de scripteurs	# total de symboles
Apprentissage	Apprentissage	122	8 982
	Validation	72	5 323
Test		85	6 341

TABLE 6.3 – Répartition des données utilisés pour l'apprentissage et le test des classifieurs de symboles isolés manuscrits en-ligne

Dans le tableau 6.4, sont données les performances des deux classifieurs présentés ici (*PMC* et *TDNN*) sur les données des sous bases rapportées dans la table 6.3.

Classifieur \ Base	Apprentissage	Validation	test		
			*Top*1	*Top*2	*Top*3
PMC	97.65%	85.01%	83.76%	92.86%	95.08%
TDNN	95.94%	86.25%	85.27%	93.83%	96.09%

TABLE 6.4 – Taux de reconnaissance des deux classifieurs (*PMC* et *TDNN*), utilisés en écrit, sur les différentes bases de la table 6.3. Le taux en *TopN* désigne le fait que la bonne classe se trouve parmi les N meilleures classes.

D'après le tableau 6.4, globalement, les résultats du *TDNN* sont significativement meilleurs en généralisation que le *PMC* (hypothèse validée par le test statistique de comparaison de proportions avec un seuil de risque de 5%). On peut noter la dégradation des performances entre l'apprentissage et la généralisation sur les bases de validation et de test. Un point intéressant, est que les performances sur la base de test augmentent sensiblement si on considère les classes reconnues en deuxième et troisième positions. On peut espérer que l'apport d'une information complémentaire, en l'occurrence provenant du signal audio, puisse permettre de faire remonter la bonne classe lorsque celle-ci se trouve classée en deuxième, voire en troisième position. Par ailleurs, une analyse plus fine, au niveau de chaque classe, montre que le comportement des deux classifieurs est très différent selon les classes. En effet, la majorité des classes ont des taux de reconnaissances très proches de 100% pour les deux classifieurs alors que pour d'autres classes, peu nombreuses, affichent de faibles taux de reconnaissance (inférieurs à 50%). Malheureusement ces deux classifieurs s'avèrent être moins performants au niveau des mêmes classes, ce qui laisse penser que la solution des problèmes rencontrés à ce niveau par leur combinaison n'apporterait pas un grand gain (comme nous le verrons plus loin). Ceci est dû au vocabulaire considéré, certaines classes sont difficilement discriminables car on se base sur le signal écrit seul, et uniquement sur les symboles isolés (*cf.* section 1.2.1).

C'est le cas par exemple des majuscules/minuscules (x et X ; y et Y), ou de certains allographes qui s'écrivent de la même façon mais dont l'interprétation dépend du contexte (x et $\times$). Dans le tableau 6.5 sont représentées les matrices de confusion (des deux classifieurs) pour les classes avec les plus faibles taux de reconnaissance et les classes avec lesquelles elles sont majoritairement confondues.

PMC/ *TDNN*[%]		Classe reconnue													
		0		o		$\times$		x		X		y		Y	
Classe attendue	0	**63.2**	**55.2**	29.9	41.4	0	0	0	0	0	0	0	0	0	0
	o	22.6	24.7	**63.4**	**63.4**	0	0	0	0	0	0	0	0	0	0
	$\times$	0	0	0	0	**50**	**40.7**	15.1	5.8	26.7	48.8	0	0	1.2	0
	x	0	0	0	0	18.9	17.6	**40.5**	**28.4**	23	41.8	0	0	2.7	1.3
	X	0	0	0	0	41.8	35.4	21.5	7.6	**34.2**	**54.4**	1.3	0	0	1.3
	y	1.2	0	0	0	0	0	0	0	2.4	1.2	**45.8**	**50.6**	38.5	32.5
	Y	0	1.2	1.2	0	2.4	1.2	1.2	0	3.6	2.4	19.9	26.2	**66.7**	**55.9**

TABLE 6.5 – Matrices de confusion partielles impliquant les classes les plus ambigües au niveau des deux classifieurs sur la base de test de la table 6.3

Le tableau 6.5 rend compte de deux points importants. Le premier concerne la confusion entre des symboles de forme géométrique proche qui ne sont pas aisés à discriminer par un classifieur traitant des symboles isolés (qu'il soit un *PMC* ou un *TDNN*). Lever cette difficulté requiert le recours à une source d'information supplémentaire : soit le contexte du symbole soit une modalité autre, comme la parole. Le second point concerne l'importance de l'évaluation des classifieurs au niveau local (des classes). En effet, même si dans le tableau 6.4 les performances globales des deux classifieurs semblent satisfaisantes, sans être parfaites, celles-ci ne reflètent pas la disparité des capacités de discrimination entre certaines classes. Lors du passage à l'échelle de l'expression complète, cela risque d'être problématique, dans la mesure ou si ces symboles apparaissent dans l'EM, cela conduira probablement à l'échec de son interprétation automatique.

6.3.2 Reconnaissance des symboles isolés parlés

De la même façon que pour le cas de l'écrit, nous découpons cette section en deux parties. La première est consacrée à la description du système et des données utilisées. La seconde rapporte les performances et leur discussion.

6.3.2.1 Présentation du système de reconnaissance de mots isolés utilisé

De façon similaire au cas des symboles isolés manuscrits en-ligne, dans le cas de l'annotation automatique des mots isolés dictés, nous sommes aussi face à un problème de reconnaissance de forme. Les trois points clés, pour le succès de cette tâche d'étiquetage, rapportés dans la section 6.3.1.1 restent valables dans le cas de la parole. En effet, il est important de disposer de données suffisamment représentatives, il est important de représenter les données dans un espace permettant une discrimination aisée des différentes classes, et enfin, il est essentiel de disposer d'un moteur de classification adapté. Dans la suite nous allons développer chacun de ces trois éléments pour converger vers la structure finale du système de reconnaissance de mots isolés utilisé dans cette étude.

Données utilisées

Dans le cas de la parole également, il est indispensable de disposer d'une base de données couvrant toutes les classes du problème. Il est aussi important que ces données soit suffisamment représentatives pour occuper au mieux les variabilités intra-classes. Cette variabilité est non seulement liée au fait que nous cherchons à mettre en place un système indépendant du locuteur, mais aussi du fait que nous considérons le cas de la parole spontanée. Dans ce contexte, les signaux de parole associés à une image de symbole présentée au locuteur afin qu'il la dicte, sont très variés. Cette variabilité due au caractère spontané de la parole vient des propriétés même de cette dernière, qui sont définies dans plusieurs travaux comme étant des disfluences [3] [PH+03, ADHB+04, HCV04, BJB+08]. Ces disfluences peuvent être des pauses (silencieuses ou sonores comme : *euh, ben, ...*), des troncations, des répétitions, des faux-départs, de l'élision [4], des hésitations mais aussi de l'agrammaticalité, etc.

À cet égard, durant la collecte de la base *HAMEX*, nous avons, fait dicter l'ensemble des symboles isolés qui la composent (74 symboles) par chacun des 58 locuteurs. De ce fait, nous disposons de 58 exemples pour chacun des symboles, ce qui fait un total de $58 \times 74 = 4\,292$ signaux (échantillons). Ces exemples sont très variés à cause de la variabilité de l'intonation [5], la variabilité dans le débit de parole et l'état émotionnel du locuteur, l'articulation variable d'un locuteur à un autre, mais aussi le genre du locuteur (41 hommes et 17 femmes), ...

La répartition des données d'apprentissage/test de la base *HAMEX* est décrite dans le tableau 5.3 pour les EMs complètes. La partie apprentissage est découpée en deux sous-parties, l'une pour l'apprentissage proprement dit (3/4) et l'autre pour la validation des paramètres (1/4). Ce dernier découpage est fait, dans le cadre de cette expérimentation, de façon aléatoire afin de disposer de la richesse en nombre

3. Les disfluences peuvent être définies comme des problèmes, des dysfonctionnements, apparaissant au cours de la parole.

4. http ://fr.wikipedia.org/wiki/Élision

5. fonctionnement signifiant des variations de la fréquence fondamentale dans l'énoncé(selon le dictionnaire encyclopédique Larousse)

de locuteurs en apprentissage et en validation. Le tableau 6.6 donne le détail de ce découpage.

Base		# de locuteurs	# total de symboles
Apprentissage	apprentissage	39	2 165
	validation		721
Test		19	1 406

TABLE 6.6 – Répartition des données utilisées pour l'apprentissage et le test des classifieurs de symboles isolés parlés

À présent nous allons présenter les traitements que subissent les signaux audio bruts avant d'être convertis dans le nouvel espace de représentation dans lequel va s'effectuer la reconnaissance proprement dite.

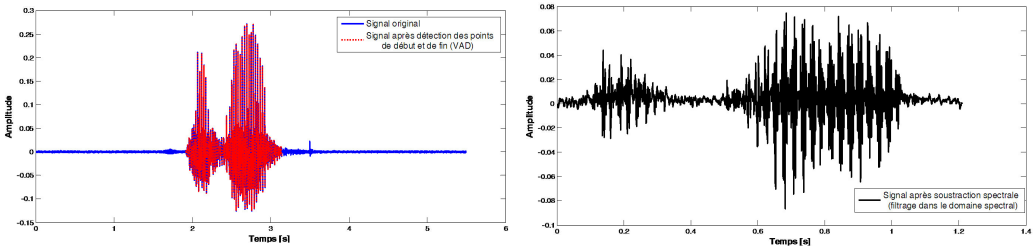
Espace de représentation des signaux de parole

Le signal audio, collecté depuis le microphone et enregistré sous forme d'un signal numérique (échantillonné à une fréquence de $48kHz$), doit, tout autant que le signal manuscrit présenté précédemment, subir des pré-traitements rendant son exploitation finale plus aisée et optimale. En effet, avant le passage à l'un des espaces de paramètres présentés dans la section 3.2.1.1 (LPC, $MFCC$, PLP), le signal de parole subit un traitement de façon globale sur l'ensemble du signal et un autre au niveau des segments élémentaires du signal total (opération sur des fenêtres qui se chevauchent) [Kol02, SRM12].

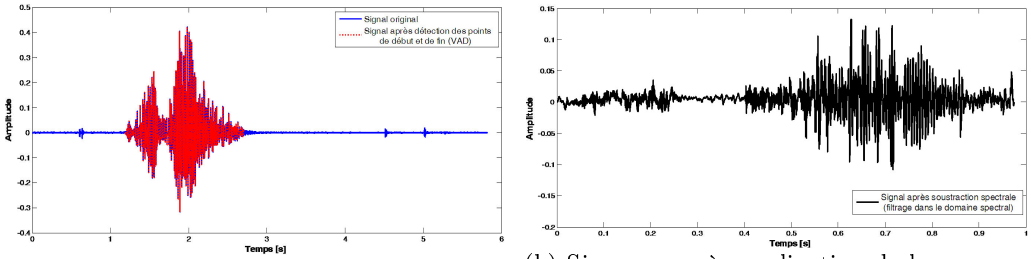
Dans notre cas, pour ce qui est du pré-traitement du signal en sa totalité, on opère dans un premier temps dans le domaine temporel en soustrayant de chacun des échantillons du signal la valeur moyenne de ce dernier. Ceci a pour but de garder uniquement la partie du signal, autour du zéro (silence absolu), qui représente effectivement la variation de la pression de l'air engendrée par le locuteur (pour conserver la dynamique du signal). Dans un second temps, une soustraction dans le domaine spectral est accomplie. Pour ce faire, deux étapes sont nécessaires. La première a pour but de détecter les zones d'activité du signal de parole. Dans le cas de la présente expérimentation où les signaux correspondent simplement à des mots isolés, cela revient à trouver une meilleure délimitation du signal de parole (chercher les temps de début et de fin de la parole effective et enlever les segments du signal correspondant à du bruit). Pour arriver à cela, l'algorithme de détection d'activité du signal (algorithme VAD, pour Voice Activity Detection), présenté par Rabiner et al. dans [RS75], est utilisé. Ce dernier est basé sur le taux de passage à zéro (ZCR en anglais pour Zero Crossing Rate) du signal ainsi que de son énergie. Après cette étape, un nouveau signal, correspondant uniquement aux zones de parole, est obtenu. Dans la deuxième phase, les zones de silence précédemment estimées, sont utilisées

pour évaluer le spectre du bruit de fond présent au moment de l'enregistrement du signal audio. Ce spectre est retranché du spectre du signal de parole effective obtenu à l'issue de la première phase, tel que c'est proposé par Boll dans [Bol79]. Sur la figure 6.6 sont rapportés deux exemples de signaux du même mot ayant subi ces deux traitements.

Locuteur 1 (femme)



Locuteur 2 (homme)



(a) Signaux initiaux (en bleu) sur lesquels sont superposés les signaux de parole effectifs détectés (en rouge)

(b) Signaux après application de la soustraction spectrale

FIGURE 6.6 – Deux exemples (de locuteurs différents) de signaux de parole correspondant au mot "*égal*", avant et après avoir subi les opérations de détection de la dynamique du signal et de suppression de bruit par filtrage spectral

Une fois ces traitements accomplis, il convient de calculer les caractéristiques qui seront extraites de ce signal. Nous utiliserons les coefficients *MFCC* (Mel-Filtered Cepstral Coefficients), ceux-ci sont présentés succinctement en section 3.2.1.1. Ce jeu de caractéristiques est l'un des plus utilisés dans la plupart des systèmes de reconnaissance actuels [HCF+06]. Ces coefficients sont intéressants puisqu'ils sont peu sensibles à la puissance acoustique du signal analysé. Ce jeu de caractéristiques est construit en considérant, en plus des coefficients cepstraux[6] eux mêmes, leurs premières et secondes dérivées. Cela, ajoute des paramètres dynamiques liés à l'évolution temporelle du signal de la parole, ce qui, d'après Furui [Fur86], améliore grandement les performances de reconnaissance.

6. Un cepstre est une transformation non linéaire d'un signal. Le cepstre d'un signal peut être, par exemple, proportionnel à la transformée de Fourier de son logarithme (http ://fr.wikipedia.org/wiki/Cepstre)

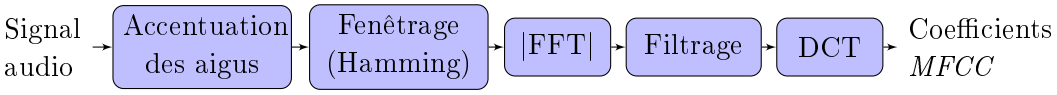


FIGURE 6.7 – Chaîne de traitement permettant l'extraction des coefficients *MFCC*

Le processus assurant le calcul de ces caractéristiques est donné en figure 6.7. Le détail de chaque étape est décrit dans la suite.

La chaîne de traitement de la figure 6.7 est réalisée sur des fenêtres de $25ms$ qui se recouvrent temporellement de 60%.
La première opération que subit chaque fenêtre du signal prétraité est l'accentuation des aigus. En effet, les sons aigus sont toujours plus faibles en intensité que les sons graves. C'est pour cela qu'un filtre passe haut est d'abord appliqué pour rehausser, en intensité, les hautes fréquences.
Le segment étant de longueur limitée, il faut l'extraire avec une fonction fenêtre. Plutôt que d'utiliser une simple fenêtre rectangulaire qui perturbe le spectre du signal, nous avons utilisé classiquement une fenêtre de *Hamming*[7] pour limiter la présence des lobes secondaires dans le domaine spectral.
Par la suite, une transformation de Fourier rapide est appliquée sur chaque fenêtre et le module est extrait et envoyé au bloc de traitement suivant. Ce dernier est un banc de filtres qui permet de caractériser le spectre précédent, et également de tenir compte du caractère non linéaire de l'audition en passant à l'échelle de *Mel* (*cf.* section 3.1).
Par la suite, une transformation en cosinus discret (DCT) est appliquée au logarithme du spectre de *Mel* obtenu précédemment. Cela permet de retrouver une représentation homogène à une représentation temporelle (application d'une transformée cosinus à un signal représenté dans le domaine fréquentiel). Ce sont les coefficients de cette dernière représentation qui sont appelés coefficients cepstraux et dénotés *MFCC*. À partir de ces coefficients sont calculées les dérivées premières et secondes qui complètent le jeu de caractéristiques que nous utilisons. Au final, 13 coefficients $MFCC$, 13 premières dérivées et 13 secondes dérivées forment les 39 éléments du vecteur acoustique (caractéristiques). Ces coefficients sont par la suite normalisés en retranchant de chacun des ordres des caractéristiques la valeur moyenne sur toutes les fenêtres pour cet ordre. Ceci a pour effet de réduire le bruit convolutionel [BKB10].

À présent, nous passons à la description de l'approche de reconnaissance de mots isolés utilisée.

Méthode de classification adoptée

Dans l'introduction du chapitre 3, nous avons déjà rapporté que le modèle statistique mis en œuvre par Jelinek et al [Jel76] reste à nos jours la base de la plupart des travaux traitant de la reconnaissance de la parole continue. Plus précisément,

7. http ://en.wikipedia.org/wiki/Window_function#Hamming_window

les systèmes basés sur des Modèles de Markov Caché (MMC) sont réputés pour être les plus performants dans la tâche de reconnaissance de la parole. Néanmoins, dans [Rab89], il est rapporté que dans le cas de mots isolés, à vocabulaire restreint, les méthodes reposant sur les MMC ont des performances équivalentes (voire inférieures dans certaines expérimentations rapportées par exemple dans [Rav10]) à celles basées sur de la mesure de similarité, qui sont les premières approches adoptées par la communauté de la reconnaissance de la parole. Dans ce deuxième cas, une base d'exemples de référence, où chaque mot est dicté par plusieurs locuteurs est utilisée. Il reste à définir une distance dans l'espace des caractéristiques et un critère de sélection parmi les exemples de référence disponibles. Dans ce cadre, la règle des $k-ppv$ (k $p$lus $p$roches $v$oisins) est bien souvent utilisée.

Le système que nous avons utilisé pour transcrire automatiquement la parole continue des EMs complètes repose sur les MMC, nous en reparlerons à l'occasion du chapitre suivant. Pour transcrire les signaux correspondants au symboles (mots) isolés, sachant que nous nous trouvons dans le cas d'un vocabulaire restreint (74 symboles), nous avons mis en œuvre un système basé sur la mesure de similarité. Les performances de ce dernier, sont en concordance avec les constatations faites par Rabiner. En effet, on arrive à de meilleurs résultats avec une mesure de similarité qu'en utilisant notre système basé sur les MMC et dédié aux EMs complètes.

Il existe énormément d'études qui portent sur des systèmes de reconnaissance de mots isolés se basant sur les coefficients $MFCC$ et une mesure de similarité utilisant une distance de déformation temporelle (DTW) [SC71, BKB10, MBE10, YJYJ11, GGY+11]. Notre système est l'implémentation de celui présenté dans [BKB10]. Pour plus de détails sur l'algorithme DTW et son application dans notre contexte, se référer à l'annexe C.

Dans notre système, une fois la mise en correspondance de l'exemple à reconnaître avec toute la base de référence effectuée, nous faisons appel à un classifieur à base de distance, un $k-ppv$. Pour un exemple à classer, une fois les k plus proches mots identifiés, le mot ayant la plus grande occurrence est attribué comme étiquette au mot à classer. Afin de disposer de scores de reconnaissance, comme dans le cas de l'écrit, nous avons calculé les occurrences de tous les mots présents parmi les k plus proches mots, que l'on a par la suite normalisées par k. Cela permet d'avoir des scores dans l'intervalle $[0,1]$, et le score d'une classe augmente avec la proportion de ses représentants. Dans le cas d'une ambiguïté entre deux ou plusieurs classes au moment de la prise de décision, un voisin supplémentaire (candidat $k+1$) est pris en compte pour trancher entre les classes. S'il n'est toujours pas possible de se prononcer en faveur d'une classe, l'étiquette du plus proche voisin est associée à l'exemple (appliquer un simple $1-ppv$).

À présent nous passons à la présentation des performances du système complet utilisé pour la transcription automatique des mots isolés dictés.

6.3.2.2 Résultats expérimentaux du système audio seul

Sur la figure 6.8 est représentée l'architecture complète utilisée dans cette expérimentation. Elle reprend les différents modules présentés précédemment (calcul des caractéristiques, base de référence, mesure de similarité, prise de décision).

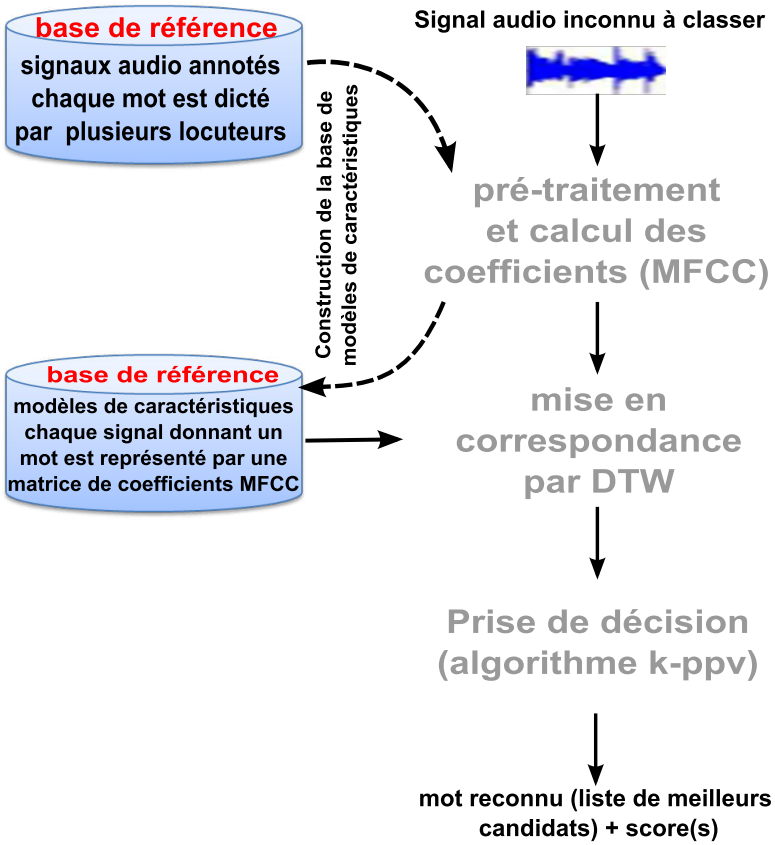


FIGURE 6.8 – Architecture globale du système utilisé pour la reconnaissance de mots isolés dictés

Les données utilisées ici sont celles décrites dans le tableau 6.6. Une fois la distance choisie, le seul paramètre de ce système concerne le choix du nombre de voisins (k). La valeur de ce paramètre est cruciale. Dans le cas d'une classification binaire par exemple, il est courant de choisir une valeur impaire de k pour éviter de se retrouver dans une situation de conflit (même nombre de voisins de chacune des deux classes). Dans notre cas, dans la mesure où on s'intéresse à avoir une liste des meilleurs candidats, le choix de $k = 1$ n'est pas envisageable, du moins pas pour le cas du système complet de fusion qu'on présentera dans la section suivante. Le choix de la valeur de k conditionne les propriétés de biais et de variance du système de décision. Une valeur faible de k limite le biais, mais au détriment d'une sensibilité peut être excessive (frontières trop découpées), à l'inverse, une valeur importante de k réduit la variance mais introduit un biais plus grand (frontières trop lisses). Dans [DHS01, CM02] est rapportée une technique heuristique, donnée

par la formule 6.6, qui permet de choisir la valeur de k de façon simple en connaissant le nombre d'observations de la base d'apprentissage (références) nb_{Obs} et le nombre de classes nb_{Classe}.

$$k \approx \sqrt{\frac{nb_{Obs}}{nb_{Classe}}} \tag{6.6}$$

Dans notre cas, le recours à la base de validation nous a permis de fixer la valeur de k à 5, ce qui minimise l'erreur de classification. Cette valeur est proche de la valeur obtenue par l'heuristique de l'équation 6.6 qui est : $\sqrt{\frac{nb_{Obs}}{nb_{Classe}}} = \sqrt{\frac{2\ 165}{74}} = 5.4$.

Nous rapportons dans le tableau 6.7 les performances globales du système sur la base de test, et cela en fonction des valeurs de k et en considérant à chaque fois le taux de reconnaissance en première, deuxième et troisième positions (s'il n y a pas suffisamment de classes parmi les k voisins pour exprimer les taux en deuxième et troisième positions, ce sont les classes suivantes présentant les distances les plus faibles qui sont prises en compte).

Valeur de k	1	3	**5**	7
Taux de reconnaissance en top 1 [%]	51.42	57.25	**60.67**	59.15
Taux de reconnaissance en top 2 [%]	71.74	79.89	**86.95**	83.31
Taux de reconnaissance en top 3 [%]	78.24	85.64	**94.85**	90.20

TABLE 6.7 – Taux de reconnaissance du classifieur audio sur la base de test de la table 6.6

Dans le tableau 6.7, on peut voir que le taux de reconnaissance en top 1 (bonne solution en première proposition) est meilleur pour $k = 5$, conformément à ce qui a été observé également sur la base de validation. On peut noter les meilleures performances de la modalité écrite (83.76% dans le pire des cas pour 60.67% ici). L'une des raisons principales est liée à la taille des bases d'apprentissage. En effet, nous disposons de plus de quatre fois plus d'échantillons en écrit par rapport à l'audio (8 982 contre 2 165) pour le même nombre de classes (74). Toutefois, dès que l'on autorise une liste étendue, le taux de reconnaissance croit rapidement pour atteindre 94.85% lorsque l'on atteint trois candidats, ce qui est comparable aux taux obtenus à l'écrit. Cela suggère qu'il est très probable qu'une source externe (l'écrit ?) en mesure de rapporter de l'information supplémentaire pourrait assez facilement faire basculer ce classement et faire passer la bonne classe en première position.

Pour conforter cette hypothèse, l'analyse de la matrice de confusion pour les classes les moins bien reconnues en audio montre que celles-ci sont différentes de celles confondues en écrit, comme l'illustre le tableau 6.8.

En conclusion, les matrices de confusion des tableaux 6.5 et 6.8, de l'écrit et de l'audio respectivement, montrent que les classes qui présentent la plus forte confusion dans la modalité manuscrite, par exemple : X et $\times$, x et $\times$ ou encore o et 0, ne le sont

Taux Reco. [%]		Classe reconnue							
		m	**n**	**l**	**e**	**o**	**i**	π	**p**
Classe attendue	**m**	**40.6**	15.6	6.3	0	0	3.1	0	0
	n	3.1	**53.1**	9.4	0	0	0	0	0
	l	15.6	12.5	**46.9**	0	0	3.1	0	0
	e	0	3.1	0	**46.9**	9.4	0	0	0
	o	0	3.1	0	9.4	**59.4**	0	0	0
	i	0	0	0	9.4	0	**50**	40.6	6.3
	π	0	0	0	3.1	0	21.9	**40.6**	12.5
	p	0	3.1	0	3.1	3.1	6.2	6.2	**59.4**

TABLE 6.8 – Matrice de confusion partielle impliquant les classes les plus ambigües au niveau du classifieur audio sur la base de test de la table 6.6

pas du tout au niveau de la modalité audio. De même, les confusions apparaissant en audio, comme par exemple entre les classes : m et n, m et l ou bien encore π et i, ne le sont pas dans le cas du signal écrit. Cette constatation est en adéquation avec les hypothèses de complémentarité énoncées tout au long de ce manuscrit.

6.3.3 Système bi-modal de reconnaissance de symboles isolés

Les résultats rapportés dans les sections 6.3.1 et 6.3.2, laissent déjà paraître la force que pourrait avoir un système ayant accès à la fois aux deux sources d'information disponibles dans les deux modalités. Dans la suite de ce chapitre, nous présentons les solutions que nous avons proposées dans un cadre bi-modal et nous finirons par donner les performances de ce nouveau système de classification bi-modal comparativement au cas mono-modal.

6.3.3.1 Présentation de l'architecture proposée

L'architecture que nous avons mise en place consiste à faire interagir les deux systèmes spécialisés (écrit et audio) présentés précédemment, grâce à l'exploitation des techniques de fusion de données décrites dans le chapitre 4.

Le système de fusion pour la reconnaissance des symboles isolés que nous avons proposé est présenté sur la figure 6.9.

La tâche des deux systèmes de reconnaissance est, pour un signal d'entrée (écrit ou audio), de proposer une liste des meilleurs candidats, ainsi que leurs scores respectifs. Ces deux listes (écrit et audio) sont par la suite fournies à l'unité de fusion

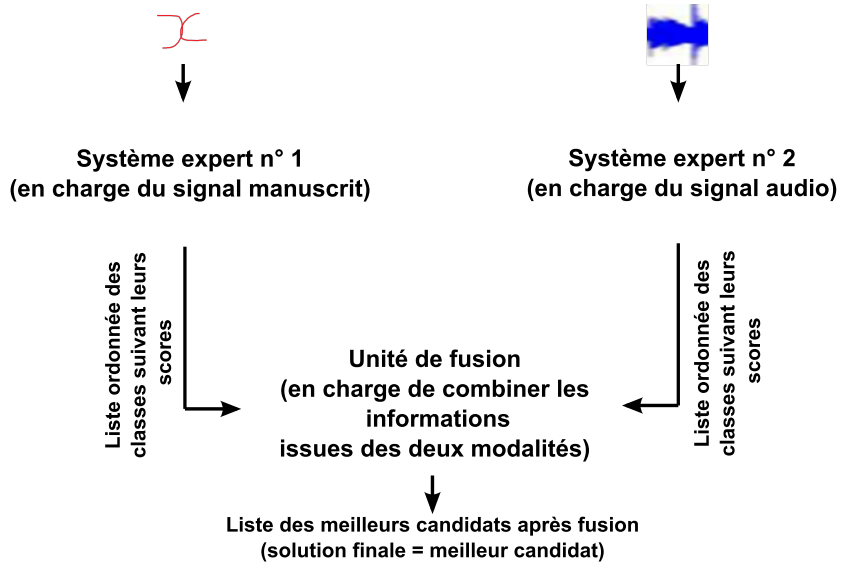


FIGURE 6.9 – Architecture du système proposé pour la reconnaissance des symboles mathématiques isolés par fusion des flux écrit et sonore

qui se charge de les combiner moyennant des techniques que nous allons présenter dans la suite. La sortie de l'unité de fusion est la classe la plus vraisemblable (ou une liste de meilleurs candidats) par rapport au couple d'hypothèses écrit/audio.

Comme nous l'avons vu au chapitre 4, de nombreuses méthodes sont disponibles pour réaliser une fusion de décisions. Typiquement, nous avons eu recours aux méthodes basées sur des règles : moyenne pondérée, règle du maximum, règle de produit, méthode de comptage de Borda ; nous avons aussi fait appel à la théorie de l'évidence en exploitant le formalisme des fonctions de croyance ; pour finir, nous avons également eu recours à une fusion basée sur des méthodes de classification, où un classifieur de type séparateurs à vaste marge (SVM) a été utilisé. Nous reviendrons, dans ce qui va suivre, sur le détail du processus de fusion impliquant chacune de ces méthodes. Avant cela, nous présentons quelques définitions utiles aux formalismes qui vont suivre.

Soit $\mathbf{C}$ l'ensemble des $Nb_{Classes}$ classes de symboles possibles, défini comme suit $\mathbf{C} = \{C_1,\ C_2, \ldots, C_{Nb_{Classes}}\}$. Une hypothèse de symbole manuscrit (x_{ecrit}), ou de symbole dicté (x_{audio}) à classer fait partie de l'ensemble $\mathbf{C}$. On définit également le score qu'une hypothèse soit de la classe C_j, par rapport à la modalité manuscrite, ou par rapport à la modalité audio ou encore par rapport à la combinaison des deux modalités, par $S_{ecrit}(C_j/x_{ecrit})$, $S_{audio}(C_j/x_{audio})$ et $S(C_j/x_{ecrit}, x_{audio})$ respectivement.

Fusion à base de règles

Nous avons dans cette catégorie exploré les six variantes de méthodes déjà définies dans la section 4.3.2.1 de manière générale. Nous allons ici les spécifier pour le

cas de notre problème, à savoir la fusion des modalités manuscrite et audio. Globalement, les N_M uplet d'hypothèses se résument dans ce cas au couple d'hypothèses (x_{ecrit}, x_{audio}).

Fusion à base de sommes pondérées

La formulation donnée par l'équation 4.4 est redéfinie comme suit :

$$S(C_j/x_{ecrit}, x_{audio}) = w_{ecrit,j} S_{ecrit}(C_j/x_{ecrit}) + w_{audio,j} S_{audio}(C_j/x_{audio}), \quad (6.7)$$

où, $w_{ecrit,j}$ et $w_{audio,j}$ sont respectivement les poids associés à la classe C_j relativement aux modalités manuscrite et audio, avec la même contrainte de normalisation que précédemment où $w_{ecrit,j} + w_{audio,j} = 1$. Selon les valeurs de ces poids, trois types de sommes pondérées ont été définies :

1 - Moyenne arithmétique : dans ce cas, quelque soit la classe C_j et pour les deux modalités, les poids sont égaux (*ie.* $w_{ecrit,j} = w_{audio,j} = 0.5$). Cela revient à dire qu'au moment de la fusion, l'avis de chacun des systèmes experts est pris en considération de la même façon. Dans cette formulation des pondérations, aucun paramètre libre n'est à régler.

2 - Moyenne pondérée par les taux de reconnaissance globaux des deux systèmes amonts : si les taux de reconnaissance globaux des deux systèmes écrit et audio sont respectivement $TauxReco_{ecrit}$ et $TauxReco_{audio}$, alors les poids de la formule de fusion de l'équation 4.4 sont formulées comme suit :

$$\begin{cases} w_{ecrit,j} = \dfrac{TauxReco_{ecrit}}{TauxReco_{ecrit} + TauxReco_{audio}}, \\ w_{audio,j} = \dfrac{TauxReco_{audio}}{TauxReco_{ecrit} + TauxReco_{audio}}. \end{cases} \quad (6.8)$$

Dans cette formulation, les deux paramètres libres sont donnés par les performances des deux systèmes spécialisés. Aucune distinction relativement aux différentes classes n'est faite, quelque soit la classe C_j, le même poids est utilisé pour chaque modalité.

3 - Moyenne pondérée par les taux de reconnaissance locaux (au niveau de chaque classe) des deux systèmes amonts : soient $TauxReco_{ecrit,j}$ et $TauxReco_{audio,j}$ les taux de reconnaissance pour la classe C_j pour les deux systèmes écrit et audio respectivement. Dans ce cas, les poids de l'équation 4.4 sont définis pour chacune des classes et pour chacune des modalités, comme

sur l'équation 6.9 :

$$\begin{cases} w_{ecrit,j} = \dfrac{TauxReco_{ecrit,j}}{TauxReco_{ecrit,j} + TauxReco_{audio,j}}, \\ w_{audio,j} = \dfrac{TauxReco_{audio,j}}{TauxReco_{ecrit,j} + TauxReco_{audio,j}}. \end{cases} \tag{6.9}$$

Cette définition des poids permet de considérer les forces de chacun des deux systèmes au niveau des classes. Dans ce cas, $2 \times Nb_{Classes}$ paramètres libres sont considérés, deux pour chaque classe (un par modalité et par classe). La prise en compte de la décision intermédiaire formulée au niveau de chaque modalité dépend de la fiabilité, pour la classe en question, des solutions proposées au niveau de cette modalité.

Fusion à base de produit [KHDM98]. L'application de cette règle, donnée par l'équation 4.5, dans notre cas se traduit par :

$$S(C_j/x_{ecrit}, x_{audio}) = 1 - ((1 - S_{ecrit}(C_j/x_{ecrit}))(1 - S_{audio}(C_j/x_{audio}))). \tag{6.10}$$

Fusion par la règle du maximum [RF03].
Ici aussi, la fusion consiste simplement à choisir le score maximal de ceux attribués aux niveau mono-modal comme donné par l'équation 6.11.

$$S(C_j/x_{ecrit}, x_{audio}) = \max(S_{ecrit}(C_j/x_{ecrit}), S_{audio}(C_j/x_{audio})). \tag{6.11}$$

Fusion à base de la méthode de Borda [dB81].
Sans reprendre la description donnée dans la section 4.3.2.1, nous allons nous focaliser ici sur la façon avec laquelle nous avons associé les voix aux classes avant fusion. En effet, comme nous l'avons précisé, il existe énormément de variantes de cette méthode, le mode opératoire que nous avons adopté est le suivant. Supposons que pour une hypothèse x_{ecrit} nous disposions de la liste de $N_{ecrit}^{meilleurs}$ candidats rangés par score décroissant. Si, de la même manière, la liste des $N_{audio}^{meilleurs}$ pour une hypothèse audio x_{audio} est aussi disponible. Alors, la liste des $N^{meilleurs}$ candidats après fusion est donnée par l'union des deux listes précédentes (tous les symboles disponibles en écrit et ceux disponibles en audio). Cela veut dire que $N^{meilleurs} = N_{ecrit}^{meilleurs} + N_{audio}^{meilleurs} - N^{ecrit \cap audio}$, où $N^{ecrit \cap audio}$ est le nombre de symboles disponibles dans les deux listes initiales (écrit et audio) à la fois. Le classement après fusion est obtenu en considérant la somme des votes pour chaque classe au niveau des deux modalités. La classe cumulant le plus de voix est le meilleur candidat associé au couple d'hypothèses (x_{ecrit}, x_{audio}) et le classement au sein de la liste après fusion est réalisé en fonction du nombre décroissant de voix. Pour ce qui est de l'assignation des votes au sein des deux modalités, celle-ci est faite en donnant $N_{ecrit}^{meilleurs}$ ($N_{audio}^{meilleurs}$ pour le cas de l'audio) voix au meilleur candidat, et en diminuant d'une unité le nombre de voix pour les classes suivantes en passant à chaque fois à la classe de rang suivant. C'est ainsi que la dernière classe de la liste se voit assigner une seule voix.

Fusion basée sur les fonctions de croyance

Nous reprenons exactement le modèle présenté dans la section 4.3.2.2. Nous allons simplement rapporter son usage dans le cadre de notre application, à savoir la fusion des flux audio et manuscrit en-ligne. Il est important de présenter la méthode adoptée pour la définition des masses allouées à chacune des classes et pour chaque modalité.

Dans notre cas, l'ensemble fondamental Ω est l'ensemble des classes $\mathbf{C}$ de notre problème.

Définition des masses de croyance dans notre cas. Comme on s'intéresse dans ce travail à combiner les modalités audio et écriture manuscrite en-ligne, les fonctions de croyance découlent immédiatement des scores de reconnaissance des deux systèmes spécialisés. Ces scores, comme évoqué auparavant, sont normalisés dans l'intervalle $[0, 1]$ et que la somme des scores des classes, pour chaque modalité, est inférieur à 1 (en multipliant par un facteur qui est le taux de reconnaissance de la modalité considérée). Si on considère la combinaison des deux hypothèses x_{ecrit} et x_{audio}, et que le processus de reconnaissance au sein de chacune des deux modalités, donne lieu à deux listes de propositions d'étiquettes (notées respectivement L_{ecrit} et L_{aaudio}) complétées par leurs scores, alors les croyances sont déduites comme suit.

Pour l'écrit, on obtient la masse de croyance :

$$m_{ecrit}(C_j) = \begin{cases} S_{ecrit}(C_j/x_{ecrit}) & si\ C_j \in L_{ecrit}, \\ 0 & sinon, \end{cases}$$

et le complément à 1, *correspondant à l'ignorance, est affecté à* $\mathbf{C}$:

$$m_{ecrit}(\mathbf{C}) = 1 - \sum_{C_j \in L_{ecrit}} m_{ecrit}(C_j).$$

De même pour le cas de l'audio :

$$m_{audio}(C_j) = \begin{cases} S_{audio}(C_j/x_{audio}) & si\ C_j \in L_{audio}, \\ 0 & sinon, \end{cases}$$

$$par\ la\ suite\ :\ m_{audio}(\mathbf{C}) = 1 - \sum_{C_j \in L_audio} m_{audio}(C_j)$$

Par la suite, ces masses sont combinées par le biais de l'équation 4.8 en prenant m_1 et m_2 comme étant respectivement les masses m_{ecrit} et m_{audio}.

Dans la suite nous donnons un exemple d'utilisation de cette approche pour fusionner les informations issues des modalités audio et manuscrite en ayant retenu 2 meilleurs candidats dans chacune d'elle.

Liste de symboles reconnus (plus leurs scores). **Exemple de masses (croyances) associées.**

$$\textit{pour } x_{ecrit} : \begin{cases} S_{ecrit}('n'/x_{ecrit}) = 0.52 \\ S_{ecrit}('x'/x_{ecrit}) = 0.46 \end{cases} \qquad \begin{cases} m_{ecrit}(\{n\}) = 0.52 \\ m_{ecrit}(\{x\}) = 0.46 \\ m_{ecrit}(\mathbf{C}) = 0.02 \end{cases}$$

$$\Rightarrow$$

$$\textit{pour } x_{audio} : \begin{cases} S_{audio}('x'/x_{audio}) = 0.62 \\ S_{audio}('s'/x_{audio}) = 0.2 \end{cases} \qquad \begin{cases} m_{audio}(\{x\}) = 0.62 \\ m_{audio}(\{s\}) = 0.1 \\ m_{audio}(\mathbf{C}) = 0.28 \end{cases}$$

Par la suite, les masses fusionnées sont données par :

$$m_{fusion}(\{x\}) = 0.46 \times 0.62 + 0.46 \times 0.28 + 0.62 \times 0.02 = 0.4264$$

$$m_{fusion}(\{n\}) = 0.52 \times 0.28 = 0.1456$$

$$m_{fusion}(\{s\}) = 0.10 \times 0.02 = 0.002$$

$$m_{fusion}(\mathbf{C}) = 0.28 \times 0.02 = 0.0056$$

$$m_{fusion}(\emptyset) = 0.52 \times 0.62 + 0.52 \times 0.1 + 0.46 \times 0.1 = 0.4204$$

Fusion à base de classification

Toutes les méthodes présentées jusque là, à l'exception de la règle de rang de Borda, sont sensibles au problème de normalisation des scores dans les deux modalités. Le but de procéder par une classification est de disposer de plus de paramètres libres. Ceux-ci seront ajustés en fonction des exemples disponibles en apprentissage pour tenir compte au mieux non seulement de la variabilité des poids à donner à chaque système expert pour chaque classe, mais également de la variabilité au sein d'une même classe.

Nous avons, pour cela, opéré une classification par SVM, connus pour leur performance très élevée [CV95]. Nous allons dans la suite en donner une brève description pour finir par présenter les propriétés du classifieur adopté dans notre architecture.

Dans notre cas, nous avons utilisé un SVM à noyau Gaussien d'écart type σ_{svm}. Cet SVM est à marge souple, sa complexité est C_{svm}. Ces deux paramètres, sont ici fixés de façon expérimentale en ayant recours à une base de validation (dont on parlera dans la section suivante, concernant les résultats).

L'apprentissage de ce classifieur est fait dans le contexte "un contre tous". C'est-à-dire que les paramètres du classifieur décrits plus haut sont appris en prenant

à chaque fois une classe pour laquelle on doit chercher les frontières relativement à toutes les autres données d'apprentissage n'ayant pas la même étiquette (classes différentes).

Les entrées considérées pour le SVM dans ce contexte de fusion sont simplement les scores alloués par chacune des deux modalités. En effet, le vecteur de caractéristiques est construit en concaténant les scores de chacune des classes du problème ($Nb_{Classes} = 74$ dans notre cas) données par les deux modalités. Cela conduit donc à un vecteur d'entrée de taille $2 \times Nb_{Classes} = 148$. La sortie de ce classifieur est également une des $Nb_{Classes}$ classes du problème.

À présent nous allons passer à la déclinaison des résultats concernant ce système bi-modal et le comparer au cas des systèmes individuels présentés auparavant.

6.3.3.2 Résultats expérimentaux du système de fusion

Les données sur lesquelles portent les résultats rapportés dans cette section sont issues de la combinaison des données des deux modalités écrit et audio des tableaux 6.3 et 6.6. La construction des exemples valides se fait, dans chacune des bases (apprentissage, validation, test), de sorte qu'un exemple de label donné de la modalité manuscrite est combiné avec tous les exemples en audio ayant le même label. Cela donne lieu à de nouvelles bases bi-modales plus variées et plus riches que celles du cas mono-modal. Dans le tableau 6.9 est résumée la répartition de ces données.

	Base d'apprentissage	**Base de validation**	**Base de test**
# d'exemples	**259 666**	**53 280**	**120 479**

TABLE 6.9 – Répartition des données bi-modales utilisées pour l'évaluation du système de fusion (combinaison des exemples écrits et audio).

D'un point de vue pratique, l'application des méthodes à base de règles sur un exemple écrit et un exemple audio à fusionner se fait sur des listes de $N^{ecrit}_{meilleures}$ hypothèses de classe en écrit et de $N^{audio}_{meilleures}$ hypothèses de classe en audio respectivement. Ces valeurs $N^{ecrit}_{meilleures}$ et $N^{audio}_{meilleures}$ sont fixées expérimentalement sur la base de validation à $N_{meilleures} = 5$. La fusion par SVM quant à elle considère la totalité des classes pour chaque modalité.

Avant de présenter les résultats concernant la fusion des modalités manuscrite et audio, nous appliquons ces techniques de fusion sur les deux classifieurs utilisés en écrit. Ceci, afin de voir l'effet de la combinaison au sein d'une même modalité (l'écriture manuscrite en ligne) avec des classifieurs qui ont une vision globale du symbole (le *PMC*) et une vision locale du symbole (le *TDNN*). Le tableau 6.10 résume les résultats des expérimentations sur la base de test écrit du tableau 6.3 où chaque exemple est d'abord reconnu par les deux classifieurs indépendamment puis les résultats sont fusionnés par les diverses méthodes exposées ici. La combinaison de ces

Systèmes évalués		Taux de reco. [%]
Rappel de la performance du **PMC** seul		**83.76**
Rappel de la performance du **TDNN** seul		**85.27**
Diverses méthodes de fusion		
Moyenne arithmétique	**simple**	**85.52**
	pondérée par les taux de reco. globaux	**85.52**
	pondérée par les taux de reco. par classe	**85.70**
Règle du produit		**85.54**
Règle du maximum		**85.52**
Règle de Borda		**85.03**
Fonctions de croyance		**85.63**
Classifieur SVM		**85.70**

TABLE 6.10 – Taux de reconnaissance de différentes méthodes de fusion des deux classifieurs du signal manuscrit (PMC et TDNN)

deux classifieurs pour la seule modalité manuscrite montre une légère amélioration du taux de reconnaissance pour toutes les méthodes de combinaison à l'exception de la méthode de Borda. Cette amélioration est très réduite relativement à la complexité du système nécessitant deux classifieurs et un module de fusion, comparée au cas d'un unique classifieur (le TDNN par exemple). Cette légère augmentation des performances tire profit des performances locales de ces deux classifieurs (*cf.* tableau 6.5). Cette expérimentation appuie le fait que certaines ambiguïtés de la modalité manuscrite nécessitent une information externe supplémentaire apportant une connaissance nouvelle qui n'est pas disponible naturellement dans le signal manuscrit. Le flux audio, tel que nous l'avons présenté dans ce rapport, apporte ce complément permettant de lever quelques unes des ambiguïtés subsistant à l'issue de la reconnaissance du tracé manuscrit.

Dans la suite nous allons présenter les différents résultats obtenus en appliquant cette fois-ci ces diverses méthodes de fusion sur les données de test du tableau 6.9 (cadre bi-modal). Le tableau 6.11 donne les taux de reconnaissance globaux (proportion du nombre de symboles bien reconnus) obtenus.

Cette fois-ci, le tableau montre que toutes les méthodes de combinaison apportent une amélioration notable au niveau de la reconnaissance relativement au cas du traitement mono-modal. Cette amélioration est différente selon les méthodes. Pour les méthodes se basant sur des règles simples (moyennes avec les différentes

Systèmes évalués		Taux de reco. [%]
Écrit (**TDNN**) seul		**85.27**
Audio seul		**60.67**
Diverses méthodes de fusion		
Moyenne arithmétique	**simple**	**89.07**
	pondérée par les taux de reco. globaux	**87.95**
	pondérée par les taux de reco. par classe	**89.14**
Règle du produit		**88.22**
Règle du maximum		**86.66**
Règle de Borda		**96.86**
Fonctions de croyance		**93.92**
Classifieur SVM		**98.04**

TABLE 6.11 – Taux de reconnaissance par fusion des modalités manuscrite et audio

pondérations, règle du produit, règle du maximum), l'amélioration est moins prononcée. En effet, le fait que ces méthodes de fusion soient exclusivement basées sur les scores de reconnaissance au niveau des deux modalités et que durant la fusion une classe est considérée à la fois (lors de la fusion, le score de la classe en cours de traitement n'est pas mis en vis-à-vis des scores des autres classes au sein de la même modalité), fait qu'elles soient étroitement liées à la différence de la dynamique des scores. Dans ces conditions, le problème de normalisation des scores est un problème sensible pour ce type d'approche. Ceci est notamment visible en considérant la combinaison de scores par la règle du maximum. Le fait de prendre l'hypothèse ayant le score le plus élevé au sein de l'une ou de l'autre des deux modalités ne garantit pas toujours le meilleur choix. Cela peut être réduit en considérant des pondérations appropriées pour chaque modalité. C'est ce que montrent par exemple les trois variantes de moyennes pondérées (simple, pondérée par les taux de reconnaissance globaux au niveau des deux modalités ou pondérée par les taux de reconnaissance par classe au niveau des deux modalités). Dans ce cas, on peut voir que prendre systématiquement la moyenne de deux scores est meilleur que de pondérer les scores au préalable par le taux de reconnaissance global au sein de chaque modalité (*cf.* équation 6.8). Ceci peut être justifié par le fait que les performances globales ne renseignent pas sur la confiance à avoir au sein d'une modalité pour une classe donnée voire pour un exemple particulier. Ceci est vérifié en considérant notamment des poids pour chaque classe, qui sont déduits à partir des performances au niveau de

chaque modalité à l'échelle de chaque classe. On constate dans ce cas, une amélioration très légère des performances par rapport à la moyenne simple, mais elle ne prend toujours pas suffisamment en compte la diversité des exemples. En considérant encre plus de poids appris sur l'ensemble de la base d'apprentissage (cas du classifieur SVM), on arrive à compenser cette différence de dynamique des scores au niveau des deux modalités. C'est cette approche qui assure la meilleure performance. Une deuxième façon de s'affranchir du problème de normalisation des scores et de ne considérer que les rangs des classes hypothèses pour chaque exemple à reconnaître. C'est ce qui est fait par la méthode de Borda et qui s'est avérée être une bonne méthode dans la mesure où elle fournit des performances proches de celles données par le classifieur SVM. Les fonctions de croyances, grâce en particulier à leur pouvoir de modélisation de l'ignorance et du conflit pour chaque couple d'hypothèses qui se présente, elles permettent également d'avoir des performances proches de celles assurées par la méthode de Borda et du classifieur SVM.

Dans les tableaux qui vont suivre, nous rapportons une analyse plus approfondie de chacune des méthodes afin de comprendre les forces et faiblesses de chacune d'entre elles, à travers la mise en évidence de leurs apports et pertes relativement aux cas mono-modaux. Ces tableaux se présentent sous la forme suivante : les nombres d'exemples bien reconnus en fusion (OK fusion) et mal reconnus (KO fusion) sont donnés pour chaque configuration initiale disponible dans les cas mono-modaux. C'est-à-dire si l'exemple est initialement bien reconnu dans les deux modalités (OK écrit, OK audio) ou pas du tout reconnu dans aucune d'elles (KO écrit, KO audio), etc.

	OK fusion		**KO fusion**	
écrit \ **audio**	**OK**	**KO**	**OK**	**KO**
OK	**66 943**	**35 681**	**0**	**107**
KO	**1 163**	**3 521**	**4 989**	**8 075**

TABLE 6.12 – Gains et pertes à l'issue de la fusion par la méthode de la moyenne simple (sur les 120479 exemples de test)

	OK fusion		**KO fusion**	
écrit \ **audio**	**OK**	**KO**	**OK**	**KO**
OK	**66 943**	**35 745**	**0**	**43**
KO	**1 062**	**2 217**	**5 090**	**9 379**

TABLE 6.13 – Gains et pertes à l'issue de la fusion par la méthode de la moyenne pondérée par les taux globaux (sur les 120 479 exemples de test)

	OK fusion		KO fusion	
écrit \ audio	OK	KO	OK	KO
OK	66 943	35 702	0	86
KO	1 166	3 580	4 986	8 016

TABLE 6.14 – Gains et pertes à l'issue de la fusion par la méthode de la moyenne pondérée par les taux au niveau des classes (sur les 120 479 exemples de test)

	OK fusion		KO fusion	
écrit \ audio	OK	KO	OK	KO
OK	66 943	35 658	0	130
KO	1 157	2 534	4 995	9 062

TABLE 6.15 – Gains et pertes à l'issue de la fusion par la règle du produit (sur les 120479 exemples de test)

	OK fusion		KO fusion	
écrit \ audio	OK	KO	OK	KO
OK	66 943	35 173	0	615
KO	2 296	0	3 856	11 596

TABLE 6.16 – Gains et pertes à l'issue de la fusion par la règle du maximum (sur les 120 479 exemples de test)

	OK fusion		KO fusion	
écrit \ audio	OK	KO	OK	KO
OK	66 943	35 186	0	602
KO	5 397	9 175	755	2 421

TABLE 6.17 – Gains et pertes à l'issue de la fusion par la règle de rang de Borda (sur les 120 479 exemples de test)

écrit \ audio	OK fusion		KO fusion	
	OK	KO	OK	KO
OK	66 943	35 634	0	154
KO	5 005	5 574	1 147	6 022

TABLE 6.18 – Gains et pertes à l'issue de la fusion par la méthode des fonctions de croyance (sur les 120 479 exemples de test)

écrit \ audio	OK fusion		KO fusion	
	OK	KO	OK	KO
OK	66 936	35 236	7	552
KO	6 007	9 939	145	1 657

TABLE 6.19 – Gains et pertes à l'issue de la fusion par la méthode de classification basée sur le SVM (sur les 120 479 exemples de test)

Cette analyse au niveau des gains et pertes orchestrés par l'opération de fusion met l'accent sur un certain nombre de points remarquables.

- Les méthodes qui agissent par des règles, que ce soit sur le score ou sur le rang, (*ie.* toutes les méthodes à l'exception du classifieur SVM) garantissent le fait que si un exemple est à là fois reconnu en écrit et en audio, ce dernier est systématiquement bien reconnu après fusion (cela est visible dans tous les tableaux, à l'exception de celui du SVM, pour lequel il existe 7 cas de "mauvaise" fusion alors que l'écrit et l'audio sont correctement reconnus). Ceci est lié au caractère de ces méthodes qui agissent soit sur le score soit sur le rang et le fait d'avoir une bonne reconnaissance au niveau des deux modalités garantit le score le plus élevé et le premier rang après fusion. Au contraire, le classifieur SVM ne prend pas chaque classe indépendamment des autres mais considère l'ensemble des scores associés à toutes les classes du problème et dans chacune des deux modalités. C'est cette configuration (qu'il prend comme caractéristiques sur ces entrées) qui permet de décider de la meilleure classe de sortie. Ce traitement permet donc de considérer de façon intrinsèque les écarts entre les scores au sein d'une même modalité mais aussi dans les deux modalités. C'est ainsi que pour le cas du classifieur SVM, 7 exemples qui sont bien reconnus à la fois en écrit et en audio ne le sont plus après fusion. Une analyse de ces exemples montre que ce sont des cas ou il y a un grand conflit entre les classes dans les deux modalités, ce qui se traduit pas une répartition presque uniforme des scores entre les classes en conflit.
- La fusion par la règle du maximum est la seule méthode qui ne permet pas de

bien reconnaître un exemple après fusion en l'ayant mal reconnu dans les deux modalités à la fois. Ceci est simplement lié à la nature de la méthode qui ne va s'intéresser qu'à retenir le meilleur score pour chaque classe, sans que ce score ne soit modifié. Ceci conduit inévitablement à mal classer un exemple qui est initialement mal classé dans les deux modalités. Les méthodes de Borda et de classification par SVM sont celles qui assure le maximum de récupération de ce type d'exemples après fusion (exemple reconnu après fusion même s'il est mal reconnu dans les cas mono-modaux). Cette force est encore une fois justifiée par le fait que ces deux approches proposent une solution pour le problème de normalisation des scores, la première en ne considérant que les rangs et la seconde en considérant non pas les scores classe par classe, mais en traitant l'ensemble des classes en même temps ce qui assure une vue de la dynamique des scores au sein d'une même modalité et relativement à l'autre modalité.

- Concernant les méthodes à base de règles (exception faite de la méthode de Borda), la fusion à base du formalisme des fonctions de croyance est celle qui assure le meilleur gain en permettant notamment de recouvrer le maximum d'exemples initialement non reconnus dans les deux modalités à la fois. Ceci est lié, comme nous l'avons déjà indiqué, à la modélisation de l'incertain (*cf.* section 6.3.3.1) dans la construction des masses à combiner. C'est d'autant plus pertinent si l'on sait que les scores dans les deux modalités lors de mauvaises reconnaissances sont souvent faibles et répartis sur plusieurs classes en laissant une grande part d'incertitude. C'est pour cela que cette façon de combiner assure des performances très proches de celles obtenues par les méthodes peu sensibles au problème de normalisation des scores (Borda et SVM). Avec les mêmes arguments, les fonctions de croyances surpassent toutes les autres méthodes à base de règles qui s'appuient sur les scores dans le cas où seul le classifieur audio permet d'avoir une bonne reconnaissance. Encore une fois, c'est en modélisant l'incertitude au niveau de la modalité manuscrite que beaucoup d'exemples mal reconnus en écrit et bien reconnus en audio sont finalement bien reconnus en fusion sans que les méthodes simples de type moyenne ne puissent arriver aux mêmes résultats.
- Un dernier point concerne le déséquilibre entre les deux classifieurs des deux modalités. En effet, pour toutes les méthodes à base de règles, très peu d'exemples reconnus seulement en écrit ne le sont pas à l'issue de la fusion, alors que ce n'est pas le cas pour les exemples reconnus uniquement au niveau de la modalité audio. Ceci traduit directement la force et la certitude des deux classifieurs spécialisés (écrit et audio). Seule la fusion à base du classifieur SVM permet de s'affranchir de ce déséquilibre.

6.4 Bilan

Dans ce chapitre, nous avons rapporté des expérimentations sur la fusion des flux sonore et manuscrit en ligne pour la reconnaissance des symboles mathématiques isolés. Le but était dans un premier temps de montrer la pertinence d'un tel traitement relativement à celui accompli dans un cadre mono-modal. Dans un second temps, nous avons présenté en détails les diverses approches de fusion que nous allons réutiliser dans le cas de la reconnaissance des expressions complètes. En effet, le résultats qui sont donnés ici montrent la meilleure performance qu'on puisse atteindre respectivement à chaque approche de fusion, car les hypothèses de symboles considérées sont garanties bien segmentées : en écrit tous les traits élémentaires appartiennent au même symbole et en audio le signal temporel correspond à la description d'un seul symbole. Dans le cas des expressions complètes, ces erreurs de classification sont accentuées par la difficulté de délimitation des symboles à classer.

Dans le chapitre qui va suivre, nous allons considérer le cas des expressions complètes et nous présenterons les différentes pistes explorées et les solutions apportées.

Chapitre 7

Système bi-modal de reconnaissance d'expressions mathématiques complètes

Sommaire

Ce dernier chapitre est dédié à la présentation du système complet de reconnaissance des expressions mathématiques complètes dans le cadre bi-modal. Les différents modules et techniques mises en œuvre pour cette tâche y sont décrits. Nous analysons également les différents résultats obtenus.

7.1 Introduction

Dans le chapitre précédent, nous avons exploré la fusion au niveau des symboles isolés. Cette expérimentation nous a permis de voir l'impact que peut avoir la combinaison des flux écrit et sonore sur la performance du système global. Toutefois, ce cas, même si il est réaliste dans le cadre d'une situation d'interaction et d'aide à la correction, ne correspond pas au traitement global d'EMs complètes. Dans cette configuration, avant d'avoir à identifier les symboles il faut d'abord être en mesure de les localiser en écrit (bons traits correspondant aux bons symboles *cf.* section 2.3) et en audio (retrouver le bon découpage du signal audio associé au bon mot *cf.* section 3.2.1.2). C'est cette problématique qui va être discutée dans ce chapitre. Dans le chapitre 6 deux niveaux de fusion ont été mis en évidence. Il s'agit, d'une part, de la fusion à l'échelle des symboles pour améliorer les performances d'identification des symboles, et d'autre part, de la fusion au niveau des relations spatiales assurant la jonction entre les symboles. Nous allons dans la suite commencer par présenter l'architecture globale proposée. Ensuite, chacun des modules la composant va être décrit.

7.2 Architecture globale proposée

De la même manière que pour le cas des symboles isolés, le type de fusion considérée est une fusion tardive. Il est de ce fait nécessaire de disposer de deux systèmes amonts experts (un pour l'écrit et un autre pour la parole). Ces derniers doivent être en mesure de fournir les décisions intermédiaires sur lesquelles va porter la fusion. La liaison entre ces deux systèmes est assurée par un module de fusion composé de plusieurs unités. La figure 7.1 donne le schéma de l'architecture proposée.

Les trois modules cités précédemment et identifiés par leurs couleurs respectives sur la figure 7.1 ont donc chacun une tâche particulière à accomplir. C'est ainsi que le module en charge de la modalité audio assure le traitement du signal correspondant à l'enregistrement de la dictée de l'EM pour fournir une (ou plusieurs) transcription(s) associée(s). Le module qui traite la modalité manuscrite se charge quant à lui de l'interprétation du signal manuscrit en-ligne pour fournir l'interprétation de l'EM. Pour finir, les modules de fusion dans cette architecture, assurent l'interaction entre les deux modalités audio et manuscrite. Ils permettent notamment d'extraire de l'information de la modalité audio pour l'introduire à différents niveaux dans la chaîne de traitement du signal manuscrit en-ligne. Dans la suite, nous allons procéder à la description des systèmes spécialisés (écrit et audio) utilisés. Ensuite nous décrirons les modules assurant la fusion et le mode de fonctionnement de notre système.

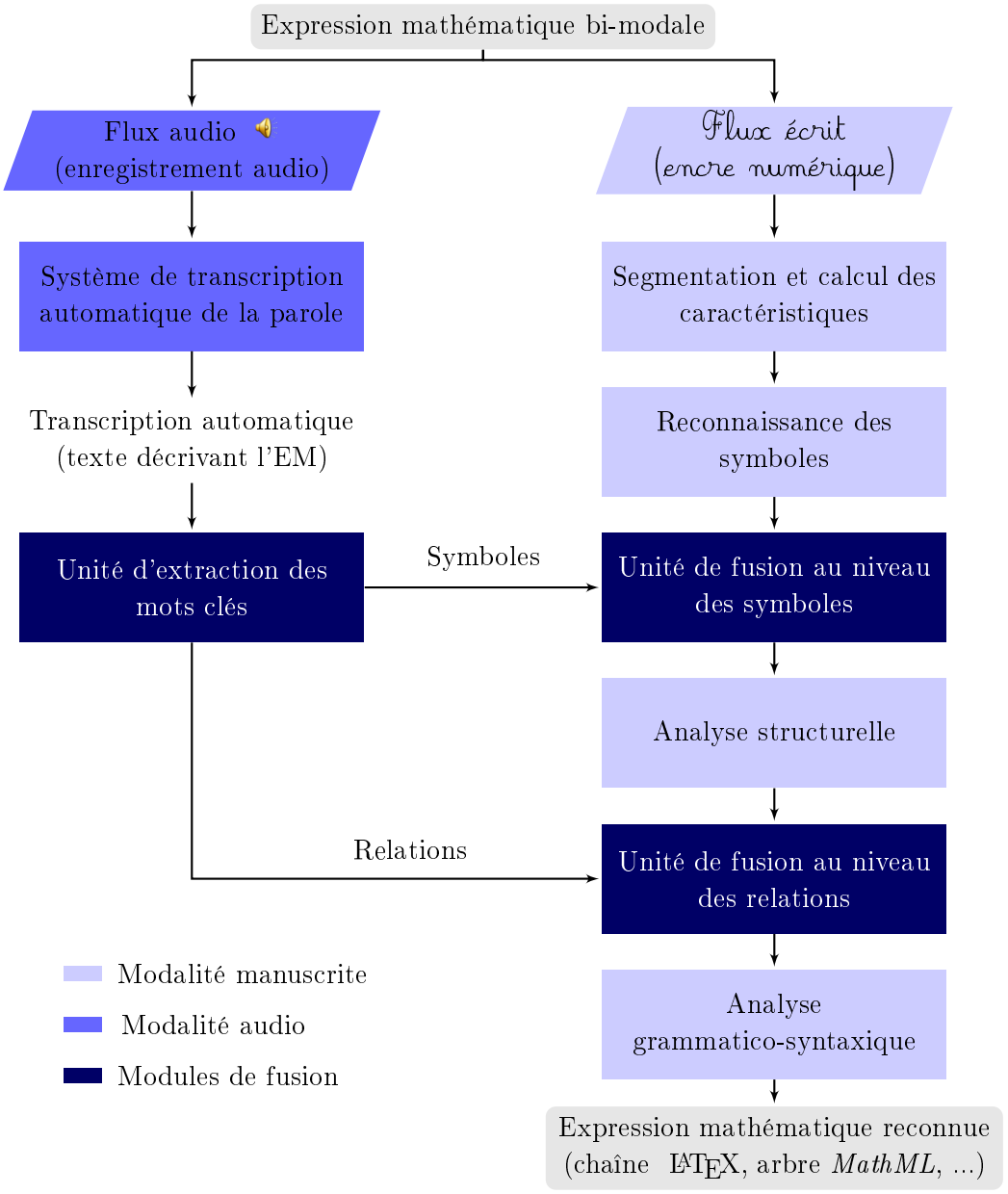


FIGURE 7.1 – Architecture globale proposée pour la reconnaissance des expressions mathématiques bi-modales

7.3 Présentation du module de reconnaissance des expressions mathématiques manuscrites en-ligne

Le problème de l'interprétation d'une EM manuscrite en-ligne peut être formulé comme étant un problème de minimisation d'une fonction de coût $Cout_{EM}$ à valeurs dans $\mathbb{R}$. Elle est définie par $Cout_{EM} : E_{traitsEM} \longmapsto \mathbb{R}$, où $E_{traitsEM}$ est l'ensemble de traits composant l'EM. Ce coût est choisi parmi ceux associés à chacune des

interprétations possibles de l'EM données par l'ensemble $E_{solutionsEM}$. Cette fonction est déduite des coûts liés aux différentes étapes composant le système global (segmentation, reconnaissance, interprétation, *cf.* chapitre 2). Il s'agit du coût associé aux scores de reconnaissance des hypothèses de symboles $Cout_{reco}$ (qualifiant les étapes de segmentation et de reconnaissance) et du coût de l'interprétation, dit aussi coût structurel, $Cout_{struct}$. Ainsi $Cout_{EM}$ est une fonction des deux coûts précédents : $Cout_{EM} = f(Cout_{reco}, Cout_{struct})$. Pour répondre à cette problématique, nous utilisons le système proposé par Awal et al. [Awa10]. L'architecture proposée s'inscrit dans la catégorie des systèmes opérant une optimisation globale des trois étapes segmentation-reconnaissance-interprétation (*cf.* section 2.6). En effet, dans ce cas, l'obtention de la solution ($Cout_{EM}$ minimal) repose sur l'optimisation simultanée de la segmentation, de la reconnaissance de symboles, et de l'interprétation. La figure 7.2 illustre cette approche telle qu'elle a été proposée par Awal et al.

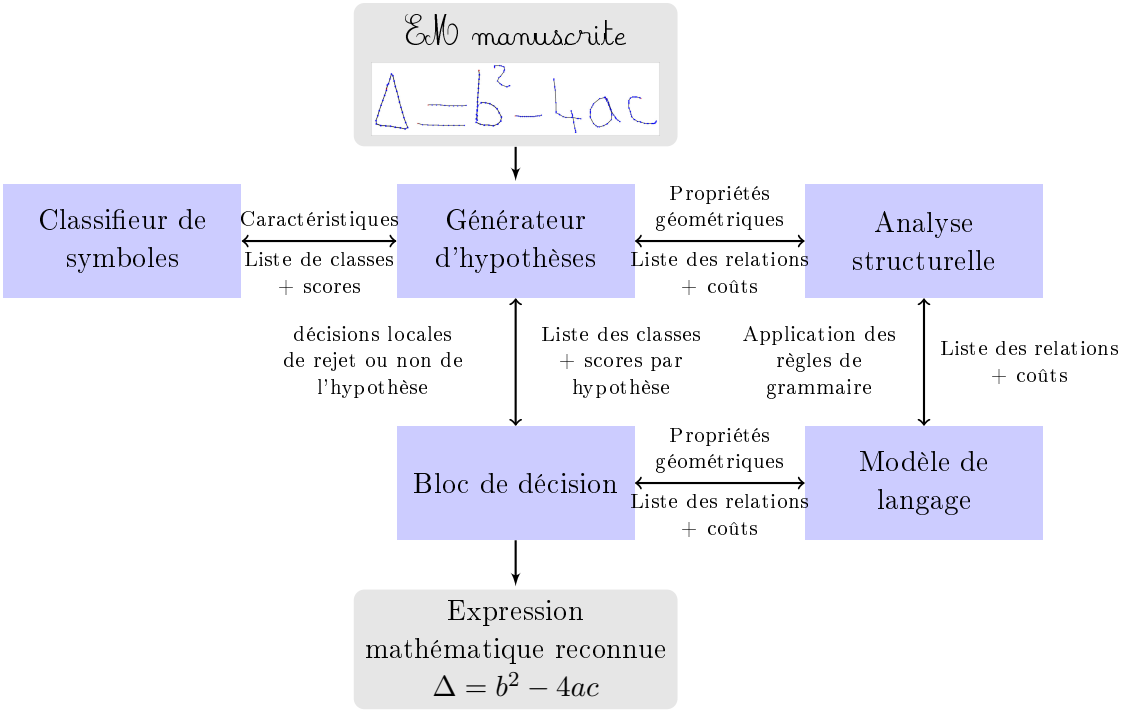


FIGURE 7.2 – Architecture du système de reconnaissance des EMs manuscrites en-ligne proposée dans [Awa10]

Dans la suite, une brève description de chaque module est donnée.

Générateur d'hypothèses

Il est en charge d'explorer les différentes hypothèses de symboles (h_s). Chaque h_s est une combinaison d'un ensemble de traits (*cf.* section 2.3). Sur l'ensemble de toutes les hypothèses de segmentation, la plupart d'entre elles sont invalides. Il peut y avoir une sur-segmentation si des traits d'un symbole sont répartis dans des

hypothèses différentes ou sous-segmentation si des traits de symboles différents sont fusionnés au sein de la même hypothèse (*cf.* figure 2.4b). Dans le cas de ce système, le générateur est basé sur une variante d'algorithme de programmation dynamique en deux dimensions ($2D - DP$). Cela lui procure la possibilité de proposer des combinaisons de traits temporellement non successifs, contrairement à une programmation dynamique $1D$. Dans le cas du signal manuscrit en-ligne en général, et des EMs en particulier, ceci est très important pour être en mesure de capturer l'information liée à l'opération de correction par des traits retardés dans le temps (ajouter les points sur des lettres, étendre les symboles élastiques tels que la barre de fraction ou la racine carrée...). En revanche cette extension $2D$ étend l'espace de recherche des hypothèses de façon exponentielle (*cf.* équation 2.1). Des contraintes de proximité géométrique et du nombre maximal de traits constituant l'hypothèse, données en section 2.3, sont utilisées pour réduire le nombre d'hypothèses sur lesquelles opérer la recherche. Ces contraintes sont :

- Le nombre de partitions, dépendant du nombre de strokes, $N_{seg}(N_{strk})$ (*cf.* équation 2.1) est limité à une borne supérieure. Celle-ci est fixée de façon expérimentale à 1500 pour éviter l'explosion combinatoire.
- Le nombre de traits par hypothèse est également limité à une borne supérieure $N_{maxTraits}$, fixée à 5 dans nos expérimentations dans la mesure où tous les symboles sont écrits au plus avec 5 traits.
- La distance entre les traits pouvant être inclus au sein de la même hypothèse est elle aussi limitée à une valeur maximale $Dist_{TraitsMax}$. Cette contrainte permet d'éliminer les groupements de traits trop éloignés.
- Le nombre de sauts autorisés dans le temps, au sein d'une même hypothèse, est de son côté borné par une valeur maximale $Nb_{SautTemp}$. Cela vient limiter le nombre de corrections qui peuvent être apportées à un symbole donné. Ce nombre, dans les expérimentations rapportées ici est fixé expérimentalement à 2.

Avec toutes ces contraintes, qui permettent de réduire l'espace de recherche et de limiter une explosion combinatoire, il est expérimentalement constaté que seulement 0.2% de vrais symboles ne sont pas proposés au moment de la génération.

Classifieur de symboles

La tâche du classifieur de symboles est la même que celle décrite au chapitre précédent pour la reconnaissance de symboles isolés (*cf.* section 6.3.1). Le comportement attendu de ce classifieur reste globalement le même, à savoir fournir une liste des meilleures classes avec leurs scores associés. La nouveauté dans ce cas est le pouvoir de **rejet** de ce classifieur. En effet, en plus des classes correspondant aux différents symboles considérés par le vocabulaire, une classe additionnelle est rajoutée. Elle correspond à un rejet d'une hypothèse de segmentation. En d'autres termes, le classifieur n'a pas tout le temps à se prononcer en faveur d'une classe correspondant aux symboles quelque soit l'exemple qui lui est présenté. Ceci était le cas des classifieurs utilisés en isolé où l'exemple à reconnaître correspondait sys-

tématiquement à une des classes du problème. Dans le cas des EMs complètes, énormément d'hypothèses de segmentation invalides sont proposées au moment de la formation des hypothèses de symbole. Ceci fait qu'un grand nombre d'hypothèses qui atteignent l'étage de classification ne correspond à aucune classe de symboles. Ces hypothèses doivent être, dans le cas de ce système, étiquetées comme des segmentations invalides grâce à la classe de rejet. Ce mode opératoire est équivalent à utiliser un classifieur hybride comme celui proposé dans [AMVG10], composé d'un classifieur ayant pour tâche de décider si une segmentation est valide ou non (classifieur à deux classes) et d'un autre dont la tâche est de se prononcer en faveur d'une des classes du vocabulaire. Toutefois, la meilleure solution retenue dans [Awa10] est l'utilisation d'un classifieur unique qui modélise en sortie les $Nb_{Classes}$ de symboles plus une classe correspondant au rejet. L'apprentissage de ce classifieur doit de ce fait se faire en même temps sur les bons exemples (vraies classes de symboles) et sur les classes correspondant aux mauvaises hypothèses de segmentations (rejet). Afin d'être en mesure d'apprendre cette classe d'exemples invalides (pas de base associée), Awal et al. ont proposé la notion d'**apprentissage global**. En effet, c'est au sein des EMs complètes et au cours de la génération des hypothèses de symboles que les exemples de la classe de rejet sont formulés (sur/sous-segmentation). Cette procédure, appliquée à la meilleure partition (segmentation de l'EM) à un instant donné, permet de garder un équilibre entre les classes de symboles valides et celle de rejet. Cet équilibre ne serait pas atteint si l'apprentissage se faisait parmi toutes les propositions d'hypothèses de symboles formulées par le générateur (risque de sur-apprentissage de la classe de rejet). Sur la figure 7.3 est montré le déroulement de cet apprentissage global pour un exemple de segmentation d'une EM à un instant donné au cours de l'apprentissage du classifieur.

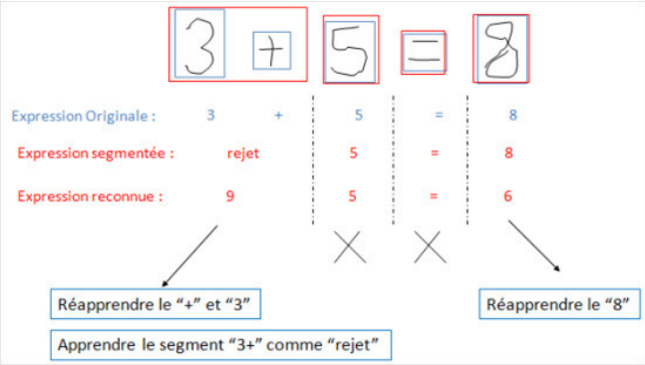


FIGURE 7.3 – Déroulement du processus d'apprentissage pour l'EM "3+5 = 8" après proposition d'une segmentation solution "95 = 6" : apprendre uniquement les exemples mal segmentés ("3", "+" et "*rejet*") et mal reconnus ("8"). Les symboles bien reconnus ("5" et "=") ne seront pas réappris [Awa10].

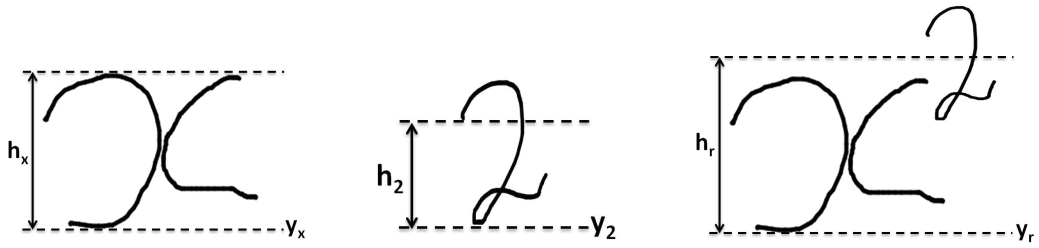
Le coût de reconnaissance d'un symbole est donné en fonction du score alloué par le classifieur pour la classe considérée :

$$Cout_{reco}(C_j/h_s) = \begin{cases} Cout_{reco}^{max}, & si\ C_j = rejet, \\ -\log\left(P(C_j/h_s)\right), & sinon. \end{cases} \tag{7.1}$$

où $Cout_{reco}^{max}$ est une constante fixée lors de l'apprentissage.

Analyse structurelle

Elle consiste à extraire les informations spatiales associées à chacune des hypothèses de symbole mais aussi associées à des sous-expressions composées de deux ou plusieurs hypothèses de symbole (dans [Awa10], le nombre d'hypothèses est égal soit à 2, 3 ou 4). Le but est de construire des coûts relationnels intermédiaires qui vont contribuer au coût structurel final ($Cout_{struct}$). Ces informations spatiales sont définies à partir des lignes de bases (pour rendre compte de l'alignement) ainsi que des hauteurs (pour rendre compte des tailles) des hypothèses de symbole. Il s'agit donc d'informations associées aux boîtes englobantes des sous-expressions, une sous-expression pouvant être une simple hypothèse de symbole (*cf.* figure 7.4).



(a) Ligne de base (y_x) et hauteur (h_x) d'un exemple de symbole de référence (pas ascendant ni descendant)

(b) Ligne de base (y_2) et hauteur (h_2) d'un exemple de digit

(c) Ligne de base (y_r) et hauteur (h_r) d'un exemple de sous-expression représentant la relation d'exposant x^2

FIGURE 7.4 – Exemples d'informations structurelles qui sont extraites [Awa10]

Dans le cas de l'exemple de la figure 7.4, la relation spatiale de la sous-expression x^2 est composée de deux éléments (x et 2). Les paramètres h et y de ces deux symboles sont estimés comme le montrent les figures 7.4b et 7.4a. Ceux de la relation x^2 sont définis comme : $y_r = y_x$ et $h_r = f(h_x, h_2)$. Dans notre cas, le coût structurel ($Cout_{struct}(R/j)$) associé à une sous-expression j pour qu'elle soit de relation R est défini en considérant l'erreur dans les positionnements relatifs des symboles dans la boîte englobante de la sous-expression par rapport à une situation idéale de la relation donnée par des règles empiriques (définies pour chaque type de relation). Ce coût est nommé coût géométrique. En annexe D sont données les différentes relations considérées par notre système de reconnaissance.

Modèle de langage (Analyse syntaxique)

L'analyse structurelle est accomplie grâce à l'utilisation d'une grammaire $2D$. Cette dernière est une combinaison de deux grammaires $1D$ dont les règles sont appliquées chacune dans une direction (l'axe horizontal et l'axe vertical). L'application successive de ces règles se fait jusqu'à atteindre les symboles élémentaires composant l'EM. Ensuite, une analyse ascendante est opérée afin de construire l'arbre relationnel de l'EM. Les règles de grammaire dans ce cas sont associées, chacune, à une relation spatiale et sont de ce fait appliquées aux entités composant ladite relation. De ce fait, au cours de la validation syntaxique, toutes les relations possibles entre ces entités sont explorées et validées par la grammaire en tenant compte des coûts relationnels. Au final, la relation choisie est celle exhibant le coût minimal parmi toutes celles qui sont valides et en considérant l'interprétation globale de l'EM.

Dans les travaux qui sont rapportés ici, nous avons utilisé différentes grammaires définies au cours des éditions 2011 et 2012 de la compétition *CROHME* sur la reconnaissance des EMs manuscrites en-ligne. Ceci est principalement motivé par deux points, le premier étant bien évidemment dû à notre participation à cette compétition. Le second point est lié au fait que la constitution de notre base bi-modale *HAMEX* n'est pas régie par une grammaire mais par une collecte des EMs de wikipédia (*cf.* chapitre 5).

Décision

C'est à ce niveau qu'est faite la recherche de l'EM solution ayant le coût global ($Cout_{EM} = f(Cout_{reco}, Cout_{struct})$) minimal. Cette solution doit être celle qui utilise tous les traits, chacun étant utilisé une et une seule fois.

Le coût total est défini de façon récursive sur les sous-expressions composant l'expression totale comme suit :

$$Cout_{EM}^{(i)} = \begin{cases} Cout_{reco}(C_j/h_s), & si\ i\ est\ un\ symbole\ (terminal), \\ \alpha Cout_{struct}(R/j) + \sum\limits_{k \in \{composantes\ de\ i\}} Cout_{EM}^{(k)}, & sinon\ (sous-expression). \end{cases} \tag{7.2}$$

Le paramètre α de cette équation sert de pondération pour exprimer l'importance du coût relationnel relativement aux coûts de reconnaissance. Consulter [Awa10] pour plus de détails à ce sujet.

7.4 Présentation du système de transcription automatique de la parole

Le comportement attendu d'un tel système, comme nous l'avons indiqué précédemment, est de produire en sa sortie une description textuelle à partir du signal de parole enregistrant la dictée d'une EM. Le système utilisé, dans notre cas, est en

mesure de fournir un graphe de mots correspondant aux différentes possibilités de découpage (segmentation) du signal audio complet. Chacun des chemins reliant les nœuds de début et de fin du signal (modélisés par des silences) donne une transcription hypothèse possible de la dictée. La solution la moins coûteuse (au regard des scores de reconnaissance donnés par le modèle acoustique et du coût de l'interprétation du point de vue du langage donné par le modèle de langage) est celle qui est retenue comme meilleure solution. C'est cette solution qui est fournie par le système comme transcription automatique du signal d'entrée. Dans ce qui suit nous présentons brièvement le détail des modules définis en figure 3.1 que nous avons utilisés pour accomplir la tâche de reconnaissance de la parole dans le cadre de notre application.

Décodeur

Le module en charge de la transcription des signaux de parole est similaire au système décrit au chapitre 3 et représenté en figure 3.1. Dans notre architecture globale, ce module est basé sur le décodeur *CMU Sphinx* [CER+07]. Ce dernier est l'un des *STAP* libres [1] les plus utilisés à travers le monde. Plus exactement, c'est *Sphinx* 3.3 qui constitue le cœur de notre *STAP*. Cette version du décodeur est celle qui fournit la meilleure précision de reconnaissance possible [RSRS00].

Ce décodeur est basé sur les Modèles de Markov Cachés continus. Ces derniers sont limités à 3 ou 5 états par phonème. Des sauts temporels dans ces modèles sont autorisés. Sur la figure 3.4 est donné un exemple de *MMC* utilisé dans *Sphinx* 3.3.

Toutefois, les ressources exploitées par ce décodeur, et fournies avec *Sphinx* 3.3, sont destinées à la transcription de la parole continue en langue anglaise. Il a fallu les adapter au cas du langage mathématique parlé en langue française.

Modèles acoustiques

Pour ce qui est des **modèles acoustiques**, nous avons eu recours à ceux mis en œuvre au sein du *LIUM* [2] [EDM+10]. Ils sont construits autour de trois points clés :

- L'unité de base modélisée (chaque modèle) est un phonème. Un mot est par conséquent modélisé par la concaténation des modèles de phonèmes. Ceci est en adéquation avec les modèles définis et utilisés par le décodeur *CMU Sphinx*.
- La modélisation d'un phonème se fait en considérant son contexte gauche et son contexte droit. Ce contexte gauche ou droit est respectivement le phonème qui précède ou qui succède le phonème de base considéré. L'ensemble (contexte gauche, phonème, contexte droit) constitue un *triphone*. Un raffinement de ces modèles est rajouté en considérant la position du phonème au sein du mot (début, milieu, fin ou encore phonème isolé).
- En langue française, il existe environ 30 phonèmes à modéliser par des *MMC*. Si les différentes configurations citées précédemment (positions du phonème au

1. cmusphinx.sourceforge.net
2. Un des laboratoires impliqués dans le projet *DEPART* finançant cette thèse.

sein du mot et ses contextes) sont considérées, il faut 324 000 états différents et 108 000 *MMC* ($30^3 \times 4$). Toutefois, toutes les combinaisons de contextes et de positions pour tous les phonèmes ne sont pas autorisées. Dans le cas des modèles acoustiques du *LIUM*, le nombre de combinaisons possibles s'élève à 9 000. De plus, face à la quantité importante des états à estimer, nécessitant par la même une quantité faramineuse de données, la notion de partage des états est exploitée. Il s'agit de factoriser les états qui se "ressemblent" en un seul état qui sera commun à plusieurs *MMC*.

Ces modèles acoustiques sont appris grâce aux outils fournis dans le projet *CMU Sphinx*. La boîte à outils *Sphinx Train* est utilisée à cette fin. L'apprentissage des modèles se fait sur des signaux de parole ayant une transcription vérité terrain. Cette transcription doit également être phonétisée. Cette description par des phonèmes de la transcription doit également être alignée de façon précise au signal de parole. Toutes ces tâches sont réalisées grâce aux outils de *Sphinx Train*.

Les corpus qui ont servi à l'apprentissage sont ceux fournis par les organisateurs de la campagne $ESTER\ 2$ [3]. Ce sont 240 heures d'enregistrements audio transcrits manuellement qui sont exploités pour l'apprentissage des modèles acoustiques.

Dictionnaire de prononciation (de phonétisation)

Dans le cas de notre application, la reconnaissance des EMs, le vocabulaire considéré est extrait du corpus audio de la base *HAMEX*. En effet, nous avons extrait l'ensemble des mots disponibles dans les transcriptions des signaux audio de la partie apprentissage de la base *HAMEX*. Chacun des 423 mots extraits est décrit par la séquence phonétique qui le compose. Les différentes prononciations possibles de chaque mot sont considérées (*cf.* tableau 7.1). Comme dans notre cas le vocabulaire est réduit, la phonétisation est faite manuellement.

Modèle de langage

Pour définir le modèle de langage relatif à notre application, nous avons utilisé les outils disponibles dans *Sphinx Train* (le CMU Statistical Language Modeling (SLM) Toolkit [4]). Il s'agit de modèles de type $n-gram$, où dans notre cas $n = 3$ (modèles $tri-gram$). Dans ce genre de modélisation stochastique du langage, l'historique du mot est représenté par les $n-1$ mots qui le précédent. Avec cette formalisation, il y a suffisamment d'information pour guider de façon efficace la tâche d'un $STAP$.

La prise en compte de séquences de mots qui seraient absentes dans le corpus d'apprentissage est assurée par l'utilisation des techniques de lissage par repli (ou *back-off*). Dans [FDM98] est donnée une description détaillée de ces modèles ainsi que des techniques de lissage associées.

Dans notre cas, les données textuelles d'apprentissage composant le corpus ayant servi à l'estimation du modèle de langage sont issues de deux sources :

3. http ://www.afcp-parole.org/camp_eval_systemes_transcription/index.html
4. http ://www.speech.cs.cmu.edu/SLM_info.html

Mots	Séquences phonétiques associées
différent	dd ii ff ai rr an dd ii ff ei rr an
deux	dd eu dd eu zz
intégrale	in tt ai gg rr aa ll in tt ei gg rr aa ll in tt ai gg rr aa ll ee in tt ei gg rr aa ll ee

TABLE 7.1 – Exemples de phonétisations possibles pour certains mots

- La première est donnée par la partie apprentissage de la base *HAMEX*. Plus précisément, nous avons utilisé le texte correspondant aux transcriptions vérités terrains des expressions mathématiques dictées d'apprentissage (2 925 *EMs*) de la base *HAMEX*.
- La deuxième source est un corpus synthétique. Ce dernier est construit en considérant environ 8 230 chaînes LaTeX d'EMs réelles extraites du web. Cela a pour but d'enrichir la base d'apprentissage en terme de variabilité des EMs. Il s'agit en fait de construire à partir des chaînes LaTeX différentes dictées synthétiques possibles pour chacune d'elles. Pour cela nous avons développé un générateur de transcriptions synthétiques *Tex2Texte*. Celui-ci génère, à partir de la description sous forme de chaîne LaTeX un arbre de type *MathML* qui est par la suite analysé (à partir du terminal le plus à gauche) pour former, de façon aléatoire, une description textuelle possible de l'EM. Comme indiqué plus haut, plusieurs descriptions possibles de chacune des EMs sont générées (par tirage aléatoire).

7.5 Présentation des modules de combinaison des deux modalités

Comme nous l'avons présenté sur la figure 7.1, le module de fusion est composé de trois unités principales. La première est directement connectée au système de transcription automatique de la parole et ne communique pas de façon directe avec le module de reconnaissance du tracé manuscrit, il s'agit de l'**unité d'extraction de mots clés**. Les deux autres unités (**unité de fusion au niveau symboles** et **unité de fusion au niveau relations**) sont quant à elles intégrées dans le système en charge de l'interprétation du tracé manuscrit. Elles sont donc en interaction

directe avec ce système. L'unité d'extraction de mots clés se charge d'extraire l'information pertinente de la modalité audio pour la transmettre aux deux unités de fusion (symboles et relations).

Dans une première approche, que nous qualifierons de **fusion par approche "sac de mots"** nous ne considérerons que la seule meilleure transcription du signal audio pour venir désambigüiser si possible le signal manuscrit. Une seconde approche, plus complète, élargira les connaissances disponibles lors de l'étape de fusion pour y rendre accessibles des informations plus riches issues du *STAP* (score de reconnaissance des mots, liste de meilleures reconnaissances, position des mots dans la transcription, ...). Dans ce cas, on parle de **fusion par alignement** et le poids des deux modalités dépend de la méthode de fusion considérée (*cf.* chapitre 6). De ce fait, le rôle des unités de fusion citées ci-haut dépend du cadre dans lequel on se place. Nous donnons dans la suite la description de chacune de ces unités en tenant compte à chaque fois du cadre de fusion (rigide ou souple).

7.5.1 Extraction des mots clés

Le vocabulaire utilisé au sein du langage mathématique regroupe deux catégories de mots. La première est la classe de mots qui sont utiles du point de vue du langage considéré (EMs). Ces mots concernent soit des *symboles*, soit des *relations*, ou encore les deux en même temps. Ils sont appelés **mots clés**. La seconde classe de mots comprend tous les autres mots. Ils ne sont nécessaires que pour donner un sens (une structure correcte), du point de vue linguistique, à la description textuelle de l'EM. Il sont identifiés par la terminologie de **mots vides**. En effet, ils n'apportent pas d'information pertinente au problème de reconnaissance des expressions mathématiques. De ce fait, la première tâche de cette première unité du module de fusion a pour rôle d'identifier les mots clés parmi tous les mots de la (ou des) description(s) textuelle(s) de l'EM. A partir de ces mots clés, la tâche suivante consiste à opérer une analyse sémantico-syntaxique (aussi bien le sens des mots que leur association sont importantes) pour extraire les symboles et relations qui y sont inclus. Ces derniers sont par la suite communiqués aux unités en charge du processus de fusion lui-même. Cette première unité du module de fusion a pour objectif de rechercher des sous-ensembles de symboles $E_{sub_{sym}}$ et de relation $E_{sub_{rel}}$ qui font partie de l'EM en cours d'analyse, et cela parmi l'ensemble de tous les symboles considérés $E_{tot_{sym}}$ et de toutes les relations $E_{tot_{rel}}$ considérées dans l'application. Ces sous-ensembles recherchés peuvent être éventuellement vides (si les deux sont vides en même temps, il y a de fortes chances que la description et/ou la transcription automatique soi(en)t erronée(s)). Cela peut être formulée par l'équation 7.3.

$$\forall\, Texte_{EM},\ \exists E_{sub_{sym}} \subset E_{tot_{sym}}\ et\ \exists E_{sub_{rel}} \subset E_{tot_{rel}}, \tag{7.3}$$

où la description textuelle (transcription automatique) de l'expression mathématique analysée est désignée par $Texte_{EM}$.

D'un point de vue pratique, la tâche d'extraction des mots clés est réalisée en opérant une analyse sémantico-syntaxique sur le texte décrivant l'EM (ou les meilleures

solutions de textes : meilleurs chemins au sein du graphe de mot). Cette analyse a pour rôle d'identifier les symboles et les relations pointés par chacun des mots clés. Il s'agit de proposer des (re)définitions de chaque mot du vocabulaire pilotant le *STAP* au sens du langage mathématique. Un mot tel que "*dix*" est par exemple redéfini comme étant la liste de symboles mathématiques "1" et "0", tandis qu'un mot comme "*donc*" ne trouve pas de définition dans notre langage cible, les EMs. Cela revient à construire un **dictionnaire** "Français-EMs" qui, à chaque mot (ou ensemble de mots) du vocabulaire associe une liste de symboles et/ou une liste de relations. Ces listes sont vides si le mot n'est pas un mot clé.

Un tel dictionnaire n'est pas disponible a-priori. Il convient donc de le construire à partir du corpus des EMs audio disponibles pour l'apprentissage. Pour chaque EM, chaque unité lexicale reconnue comme un mot clé est traduite en un équivalent de un ou plusieurs symboles et/ou relations. Sa construction est accomplie en opérant la même analyse syntaxique de tout le corpus d'apprentissage de la base *HAMEX* (transcription manuelle ou vérité terrain des signaux audio). Sur la figure 7.5 est montré le processus de construction du dictionnaire en question et un exemple de son utilisation à des fins de reconnaissance.

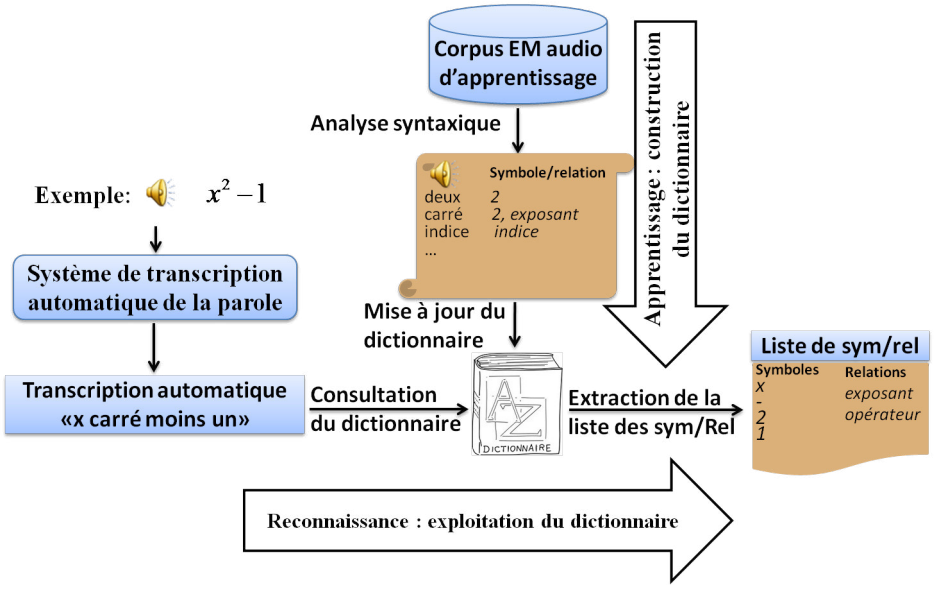


FIGURE 7.5 – Processus d'extraction des mots clés : mise à jour du dictionnaire en fonction du corpus disponible en apprentissage, puis exploitation du dictionnaire.

Suivant que l'on considère une fusion par approche "sac de mots" ou une fusion par alignement , l'extraction de mots clés ne se fait pas de la même façon. Dans le premier cas, comme nous allons le voir par la suite, seule l'information de présence des symboles/relations est importante. Il s'agit d'extraire un **sac de symboles/-relations** clés qui sera disponible dans la description textuelle de l'EM en cours de traitement. Pour le cas de la fusion par alignement , en plus de l'identification des mots clés, les informations de score de reconnaissance et de position temporelle des

mots au sein du signal total décrivant l'EM sont importantes. Cela veut dire que si par exemple un mot clé est présent plusieurs fois au sein de l'EM et à des localisations temporelles différentes dans le signal, dans le cas de la fusion par approche "sac de mots" le mot est considéré une seule fois (il est présent), abstraction faite des différentes positions auxquelles il se trouve, tandis qu'en fusion par alignement chacune des occurrences est considérée et complétée par les informations citées plus haut. À titre d'exemple, si l'on obtenait la transcription "*neuf cent quatre-vingt-dix-neuf*", l'analyse de celle-ci donne lieu à un seul symbole : le "9", lorsqu'on est en fusion par approche "sac de mots", alors que si c'est pour accomplir une fusion par alignement , il est nécessaire d'extraire les trois occurrences du symbole "9" en spécifiant leurs scores de reconnaissance respectifs et leurs ordre d'apparition (l'information du temps de début et de la durée temporelle).

A ce stade, les informations extraites de la transcription automatique du signal audio se présentent sous la forme d'une double liste de symboles et relations, celles-ci sont exploitées par les unités de fusion associées. Dans la suite nous exposons ces unités en question.

Nous commençons par présenter les approches mises en place pour la tâche de fusion au niveau des symboles. Nous l'avons laissé entendre plus haut, nous avons utilisé deux approches de fusion (au niveau des symboles) dépendant de la façon de considérer l'influence de la modalité audio : une configuration où l'on considère que l'information donnée par la modalité audio est limitée à la seule meilleure transcription, on l'a nommée "*fusion par approche sac de mots*" ; une seconde configuration où la combinaison des deux modalités se fait en tenant compte d'information de plus bas niveau, on l'appelle "*fusion par alignement* ". Dans la suite nous allons présenter chacune de ces deux approches.

7.5.2 Fusion d'information au niveau symboles par approche "sac de mots"

Dans ce cas, l'information additionnelle apportée par la modalité audio est considérée sûre (équivalent à dire que le score de reconnaissance de chaque mot est de 1). De la modalité audio, seules les étiquettes des propositions de symbole sont importantes et on fait abstraction de la certitude du système audio (scores des symboles et listes des hypothèses de reconnaissance en conflit).

Dans ce cadre, on propose d'appliquer un ré-ordonnancement de chaque liste des hypothèses de classe donnée par le classifieur des hypothèses de symbole formulées au niveau de la modalité manuscrite. Ce ré-ordonnancement résulte de la modification du score attribué à chaque classe. Cette modification vise à pénaliser une classe si celle-ci est absente dans la liste des symboles retenus en audio ; ou bien au contraire à consolider la proposition faite par la modalité manuscrite dans le cas où la classe est disponible dans les deux modalités. La schématisation d'un tel scénario est donnée sur la figure 7.6.

Comme cette figure le montre, l'écrit fournit une liste ordonnée des meilleures classes $C_1^{ecrit} \dots C_n^{ecrit}$ dans l'ordre décroissant de leur score. À l'issue de la fusion,

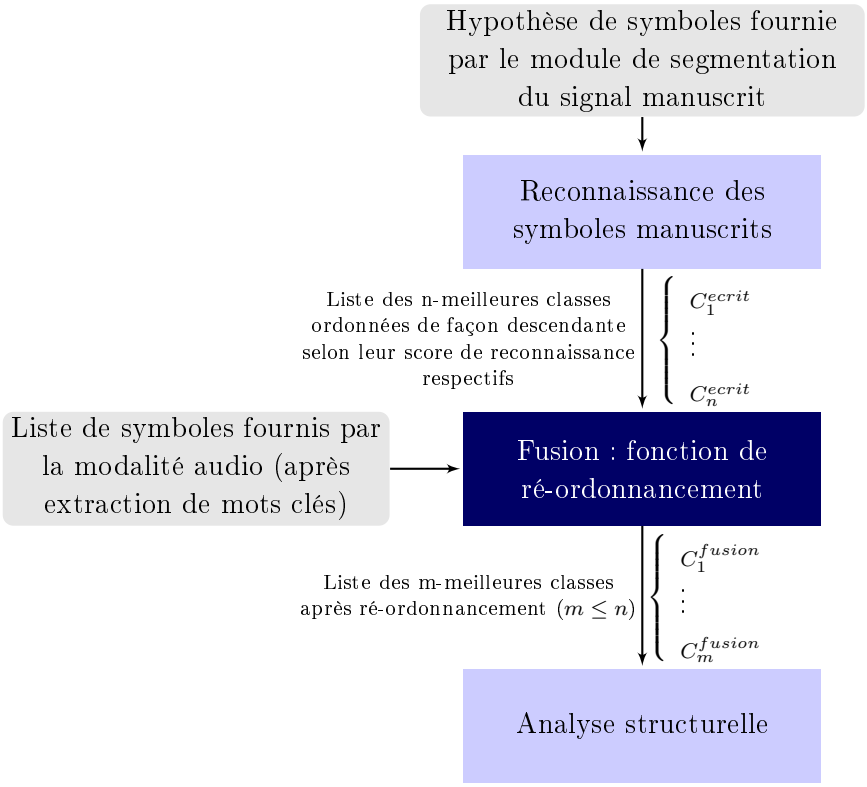


FIGURE 7.6 – Schéma de principe de la fusion à base de fonctions de ré-ordonnancement (fusion par approche "sac de mots")

cette liste peut subir un changement dans l'ordre des classes, et de plus certaines classes peuvent être filtrées pour obtenir une nouvelle liste $C_1^{fusion} \dots C_m^{fusion}$. Cela dépend de la règle de fusion utilisée.

Trois types de règles sont mises en œuvre pour réaliser la tâche décrite ici. Ces règles sont soit une transformation linéaire, soit une transformation sigmoïdale ou bien un filtrage par seuillage. Dans la suite est donnée une brève description de chacune d'entre elles.

7.5.2.1 Fusion par seuillage.

C'est la méthode de fusion la plus stricte dans la mesure où les classes de symboles retenues pour chacune des hypothèses formulées en écrit ne seront maintenues pour la suite du traitement que si elles sont disponibles à la fois en écrit et en audio ou que le système écrit est suffisamment sûr de cette solution (score en écrit supérieur à un seuil à définir). En d'autres termes une hypothèse de classe est viable pour la suite de traitement si les deux modalités sont d'accord ou que l'expert écrit est certain de sa proposition (nécessité d'avoir un classifieur écrit assez fiable). Cette

méthode de fusion peut être formulée par l'équation 7.4.

$$S(C_j/x_{ecrit}, lSymAudio) = \begin{cases} S_{ecrit}(C_j/x_{ecrit}) & si\ C_j \in lSymAudio\ \ ou \\ & S_{ecrit}(C_j/x_{ecrit}) > S_{seuil}, \\ 0 & sinon, \end{cases} \quad (7.4)$$

avec :

- $S_{ecrit}(C_j/x_{ecrit})$ est le score avant fusion pour que l'hypothèse de symbole formulée en écrit x_{ecrit} soit de classe C_j,
- $S(C_j/x_{ecrit}, lSymAudio)$ est le score après fusion pour l'hypothèse de symbole x_{ecrit} connaissant la liste de symboles audio $lSymAudio$,
- S_{seuil} est le seuil sur les scores exprimés par l'écrit à partir duquel la classe proposée est maintenue pour la suite du traitement même si elle n'est pas présente en audio. Dans nos expérimentations, celui-ci est fixé de façon expérimentale sur une base d'apprentissage.

C'est dans ce cas, qu'après fusion, la liste des étiquettes pour une hypothèse écrit donnée peut être de taille inférieure ($m \leq n$, *cf.* figure 7.6). En revanche, si le rang peut changer en fonction du retrait ou non de certaines classes, les scores restent les mêmes que ceux formulés par le classifieur des symboles en écrit (l'ordre au sein des classes restantes reste le même).

7.5.2.2 Fusion par transformation linéaire.

Cette approche se veut moins stricte. Une hypothèse de symbole obtenue à l'écrit, même si son score est inférieur au seuil défini plus haut (S_{seuil}) et si elle est absente dans la liste de symboles donnée par l'audio ($lSymAudio$), a la possibilité d'être maintenue pour la suite du traitement. Toutefois, si la liste après fusion contient toutes les classes disponibles en écrit ($m = n$, *cf.* figure 7.6), le classement peut évoluer selon leur présence ou absence dans la liste $lSymAudio$. Ceci est donné par la formule 7.5 :

$$S(C_j/x_{ecrit}, lSymAudio) = \begin{cases} \min[1, \alpha_r S_{ecrit}(C_j/x_{ecrit}) + \beta_r] & si\ C_j \in lSymAudio \\ \min[1, \alpha_p S_{ecrit}(C_j/x_{ecrit}) + \beta_p] & sinon, \end{cases} \quad (7.5)$$

avec :

- α_r et α_p sont les pentes des fonctions de fusion linéaires traitant respectivement les symboles présents (rehaussement) et absents (pénalisation) en audio,
- β_r et β_p, quant à eux, sont les points (ordonnées) à l'origine des fonctions linéaires de fusion traitant respectivement les symboles présents et absents en audio.

Le caractère linéaire de la transformation, fait que la même modification des scores est appliquée quelque soit la valeur de ce dernier (que le classifieur écrit soit

très sûr ou non). Seule l'information de présence ou non dans la liste fournie en audio permet de choisir d'appliquer une conversion différente du score (score après fusion plus élevé que celui de l'écrit seul où inversement).

7.5.2.3 Fusion par transformation sigmoïdale.

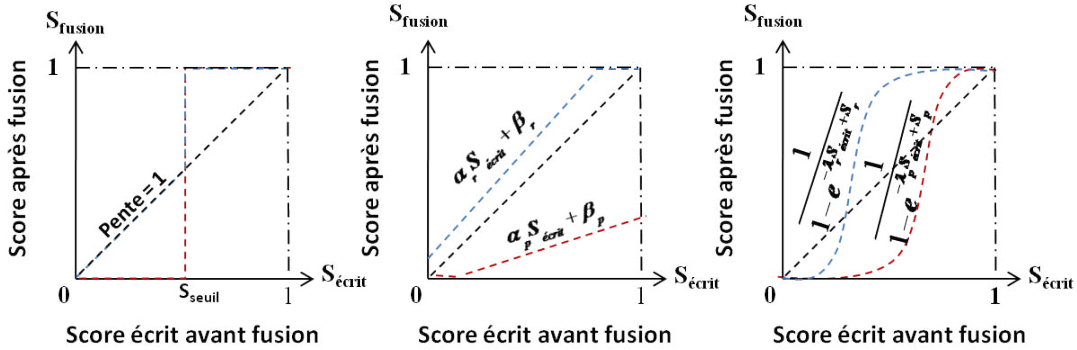
Pour tenir compte de la précision du classifieur écrit, nous proposons de garder la même formalisation précédente mais en substituant des fonctions logistiques aux transformations linéaires. Plus exactement, ce sont des transformations sigmoïdales qui sont appliquées. Dans ce cas, la transformation appliquée au score initial dépend de sa valeur : plus il est élevé, plus le score final sera important et inversement. Ceci est formulé par :

$$S(C_j/x_{ecrit}, lSymAudio) = \begin{cases} \dfrac{1}{1-e^{-\lambda_r \times S_{ecrit}(C_j/x_{ecrit})+S_r}} & si\ C_j \in lSymAudio \\ \dfrac{1}{1-e^{-\lambda_p \times S_{ecrit}(C_j/x_{ecrit})+S_p}} & sinon, \end{cases} \tag{7.6}$$

avec :

- λ_r et λ_p sont respectivement les pentes des fonctions sigmoïdes pour les symboles présents et absents en audio.
- S_r et S_p, quant à eux, représentent les centres de ces sigmoïdes.

Le choix de ces trois méthodes est justifié par le fait que celles-ci se comporte différemment face au problème de la dynamique des scores (très forts scores et faibles scores à l'écrit). Les actions de chacune de ces trois approches de combinaison peuvent être représentées schématiquement comme le montre la figure 7.7.



(a) Fusion à base de seuillage

(b) Fusion à base de transformation linéaire

(c) Fusion à base de transformation sigmoïdale

FIGURE 7.7 – Différentes approches de fusion par approche "sac de mots" : en rouge les transformations de pénalisation appliquées lorsque la classe n'est pas disponible en audio. En bleu sont représentées les transformations appliquées lorsque la classe est disponible au niveau des deux modalités à la fois.

Dans ce genre de méthodes, bien que l'information audio (liste des symboles) soit considérée comme complètement fiable, une classe de symbole présente en audio et qui n'apparaît pas du tout en écrit ne sera jamais considérée pour la suite du traitement. Toutefois, cela est peu gênant si l'on considère que le classifieur est capable de proposer la véritable étiquette en seconde voire en troisième position. En effet, c'est ce que l'on a constaté lors de l'évaluation des classifieurs écrit à l'occasion du chapitre précédent concernant la reconnaissance des symboles isolés (le taux de reconnaissance des symboles du $TDNN$ en première position est de $85,27\%$ et il est de $96,05\%$ si la liste des trois meilleures reconnaissances est considérée). C'est cette observation qui a motivé le recours à des méthodes de fusion plus souples.

7.5.3 Fusion d'information au niveau symboles par alignement

Dans ce cadre, nous cherchons à favoriser une mise en correspondance des modalités qui tienne compte de leur alignement temporel. En effet, il n'est plus question de considérer le sac de symboles extrait de la transcription automatique du signal audio (ou du treillis de mots associé), mais d'avoir un découpage du signal complet en segments. Chaque segment représente une hypothèse (ou une liste d'hypothèses) de mot(s) de laquelle est extraite l'hypothèse (ou la liste des hypothèses) de symbole(s). L'objectif est par la suite de trouver les associations groupement écrit (hypothèse de segmentation en écrit : regroupement de traits) et segment audio associé (hypothèse de segmentation en audio : portion du signal audio). L'association n'est valide que si le groupement écrit et le segment audio impliqués représentent la même information (même symbole mathématique dans ce cas). La finalité de cette association est d'être en mesure, une fois cette étape accomplie, d'appliquer toutes les techniques de fusion définies dans le chapitre précédent concernant la fusion au niveau des symboles isolés. Pour illustrer ce propos, nous donnons sur la figure 7.8 un exemple de partitions de l'écrit (choix des traits faisant partie d'une même hypothèse de symbole) et de l'audio (choix des frontières entre les portions de signal donnant chacune une hypothèse de mot) et l'association souhaitée entre elles.

Chaque groupement écrit (resp. chaque segment audio) est identifié pas des informations d'étiquette, de score et de positionnement relativement au signal complet.

Contrairement au cas de la fusion par "sac de mots" où un regroupement de traits conservait comme étiquettes possibles seulement celles présentes initialement (fournies par le classifieur écrit) avec des scores réévalués en tenant compte du contenu du sac de mots, ici toutes les étiquettes provenant de l'audio vont pouvoir apparaitre et être associées à un regroupement de traits.

Si dans le cas de l'écrit les hypothèses de symboles conduisant à la formulation des différentes partitions possibles sont construites au cours de l'analyse de l'EM (*cf.* 7.3), dans le cas de la modalité audio, nous disposons, à l'issue de la transcription du mot d'un treillis de mots. Ce dernier est défini par :

- Chacun de ces nœuds représente le mot associé à une portion du signal complet. C'est le **segment** de mots identifié précédemment.
- Un segment est identifié par : sa localisation, son étiquette et un score associé.

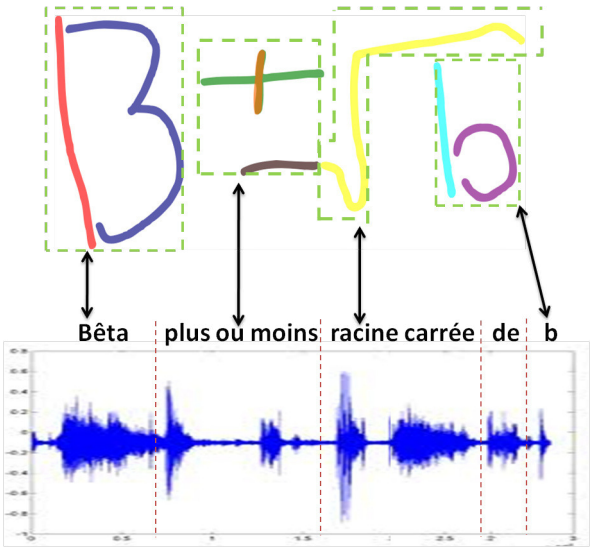


FIGURE 7.8 – Exemple d'association groupements/segments pour deux partitions écrit et audio données et représentant la même EM

On donnera dans la suite plus de détails à ce sujet.

- Tous les chemins de ce graphe de mots représentent tous les découpages envisageables du signal complet (partitions audio). Chacun de ces chemins est sanctionné par un coût issu des scores acoustiques (de reconnaissance) alloués par le système audio. Le meilleur chemin de ce treillis donne la solution proposée par le système audio. Il s'agit de la transcription automatique (le texte) proposée par le système audio pour interpréter le signal. Celle-ci est obtenue en considérant les coûts des chemins issus des scores acoustiques mais aussi les scores attribués par le modèle de langage pour cette interprétation.
- À chaque mot de la meilleure solution est également associé un score (donné par la combinaison du score de reconnaissance donné par le modèle acoustique et du score linguistique donné par le modèle de langage).

Sur la figure 7.9 est donné le graphe de mots résultant de la reconnaissance de l'EM "$x = 50$" par le système de TAP. La vérité terrain fournie par la dictée du locuteur est "*x est égal cinquante*" et la transcription automatique donnée par le STAP pour cet exemple simple correspond à la vérité terrain.

Dans l'exemple de la figure 7.8 sur l'association groupements/segments, où seule la meilleure transcription du signal audio est considérée, les segments audio considérés ne sont pas forcément composés que d'un seul mot (1 mot = 1 nœud du treillis). En effet, comme on peut le voir, pour le cas du symbole "$\pm$", trois mots sont mis en commun pour former le segment associé. Il en est de même pour le cas du symbole "$\sqrt{}$". Il convient effectivement de ne pas utiliser directement les mots du treillis au moment de la fusion mais leur équivalent en langage mathématique (*cf.* section 7.5.1). Cette considération rend la définition des segments en audio au niveau symboles indispensable pour être en mesure d'aligner les informations venant

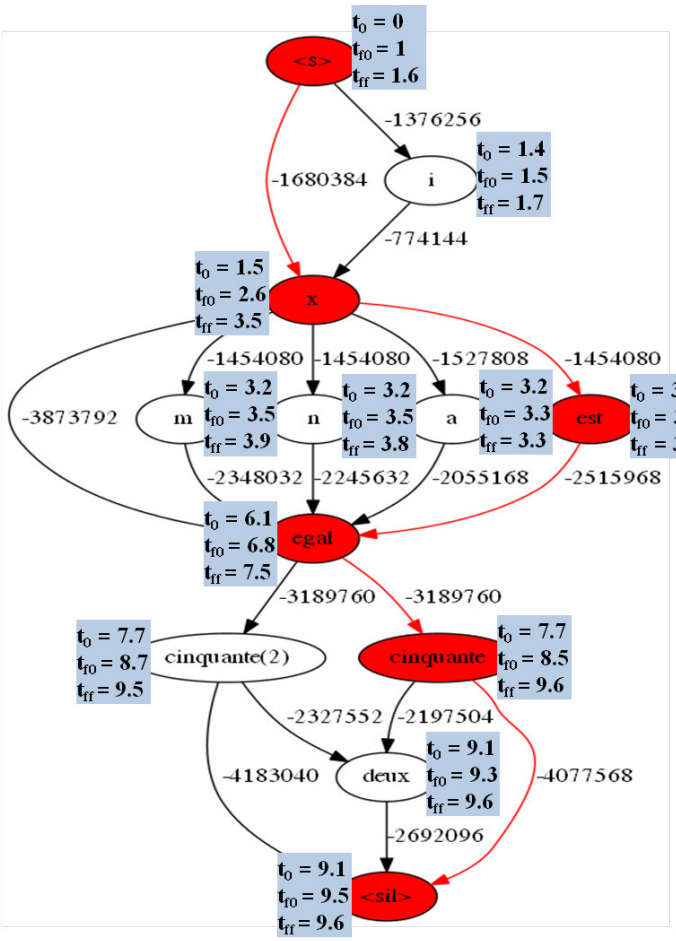


FIGURE 7.9 – Exemple de treillis de mots fourni par le système de TAP : t_0 est le temps de début du segment donnant l'hypothèse ; t_{f0} est le temps de début de l'intervalle de temps où se situe le temps de fin du segment ; t_{ff} est le temps de fin de l'intervalle de temps où se situe le temps de fin du segment. Les poids des arcs représentent le score acoustique (de reconnaissance) du mot se trouvant sur le nœud de destination. Les balises "<s>" et "<sil>" représentent respectivement les silences avant et après la dictée, tout chemin valide doit assurer le lien entre eux.

des deux modalités. Dans la suite, nous allons présenter le processus adopté pour la construction des hypothèses de segments au niveau de la modalité audio.

7.5.3.1 Définition du segment audio.

À la lumière de ce qui a été rapporté plus haut, deux cas sont à distinguer : si on considère la transcription automatique fournie à la fin du traitement ou si on se place en amont dans la chaîne de reconnaissance de la parole et on considère alors le graphe de mots à partir duquel est extraite la meilleure transcription finale. Le découpage en segments est plus direct au niveau du texte de la meilleur transcription

et les scores donnés dans ce cas, tiennent à la fois compte des modèles acoustiques (reconnaissance des mots) et du modèle de langage (validité linguistique du texte). Dans le cas du treillis complet, les frontières ne sont pas clairement définies entre les segments : les temps de fin des segments sont donnés par des intervalles ce qui leur permet de se connecter à des segments suivants ayant des temps de début différents. Nous allons dans la suite présenter la définition des segments selon ces deux points de vue.

A. Définition du segment en considérant la meilleure transcription audio. Si on considère la meilleure description textuelle, le découpage en segments est déduit à partir de la liste des mots (L_{mots}) composant le texte (mots clés et mots vides). Comme introduit plus haut, chaque mot se voit associé un score (déduit de son score acoustique et de celui du langage) et une localisation temporelle précise (temps de début et durée). On peut ainsi pour chaque élément de L_{mots} former un quadruplet de la forme $(mot, score, tempsDebut, tempsFin)$. Les segments correspondent dans ce cas aux symboles extraits de l'analyse syntaxique de cette liste de mots. Chaque segment, ainsi défini, contient une seule hypothèse de symbole (noté $[(mot, score, tempsDebut, tempsFin)]$) qui est la meilleure classe associée à la portion du signal considérée. À l'issue de l'analyse syntaxique, une nouvelle liste L_{Sym}, de symboles cette fois-ci, est déduite. Elle a la même structure que L_{mots} (un élément est un quadruplet), sauf qu'elle représente les informations liées à chacun des symboles mathématiques déduits des mots $((ClassSym, score, tempsDebut, tempsFin))$. La taille de L_{Sym} peut être différente de celle de L_{mots}. En effet, un symbole peut provenir de plusieurs mots ("est égal" $\rightarrow$ '=') ou inversement, un mot peut représenter plusieurs symboles ("cinquante" $\rightarrow$ '5', '0').

L'exemple précédent est représenté par la liste de mots suivante : $L_{mots} = \{[(x, 1, 1.5, 3.2)]; [(est, 0.86, 3.2, 3.7)]; [(egal, 1, 3.7, 7.7)]; [(cinquante, 0.63, 7.7, 9.4)]\}$. La liste des segments en symboles est alors : $L_{Sym} = \{[('x', 1, 1.5, 3.2)]; [('=', 1, 3.7, 7.7)]; [('5', 0.63, 7.7, 8.5)]; [('0', 0.63, 8.5, 9.4)]\}$. Dans cet exemple le mot "cinquante" représente deux symboles mathématiques ($'5'$ et $'0'$). Lors de l'analyse syntaxique, deux segments sont générés. Ils auront pour score le même que celui du mot associé et ils partageront la plage temporelle associée au mot à parts égales *dans l'ordre naturel dans lequel ils auraient été tracés au niveau de la modalité manuscrite.* Dans l'exemple du "cinquante", en admettant une écriture de gauche à droite, un scripteur ordinaire écrira $'5'$ puis $'0'$. Dans le cas de la fusion de mots pour former un ou plusieurs symbole(s), le score associé au(x) symbole(s) est le score moyen des mots fusionnés et la plage de temps des mots en question est répartie sur l'ensemble des symboles suivant le même principe que précédemment.

B. Définition du segment en considérant le treillis de mots. Dans le cas où le graphe de mots est utilisé pour construire les segments, deux étapes sont mises en œuvre. La première concerne le regroupement de toutes les hypothèses de mots en conflit au sein du même segment (de mots). Par la suite chaque segment

de mots subit le même traitement que précédemment pour créer les segments au niveau symboles/relations.

Pour ce qui est du traitement des mots en conflit (en concurrence pour l'interprétation d'une même portion du signal audio complet) à partir du graphe, ce sont les temps de début qui définissent les différents segments de mots. Les mots correspondant à des portions du signal ayant le même temps de début sont considérés comme les différentes hypothèses de mot pour un même segment. C'est ainsi que sur la figure 7.9, la liste des segments de mots est donnée par : $L_{mots} = \{[(i, -1376256, 1.4, 1.5, 1.7)]; [(x, -1680384, 1.5, 2.6, 3.5)]; [(m, -1454080, 3.2, 3.5, 3.9); (n, -1454080, 3.2, 3.5, 3.8); (a, -1452032, 3.2, 3.3, 3.3); (est, -1452032, 3.2, 3.4, 3.5)]; \ldots\}$. Dans ce cas, comme nous l'avons déjà précisé, pour pouvoir lier un nœud du treillis à ceux qui le suivent temporellement et qui commencent à des instants différents, le temps de fin n'est pas une valeur fixe mais un intervalle. De ce fait chaque hypothèse de mots est définie par le quintuplet suivant :
$(mot, score, tempsDebut, BorneInftempsFin, BorneSuptempsFin)$.

Chaque segment dans ce cas contient différentes hypothèses de mots qu'il faudra analyser sémantiquement et syntaxiquement comme nous l'avons expliqué précédemment pour déduire les segments des symboles. Dans ce cas, la difficulté est plus grande que dans le premier cas (considération de la description textuelle uniquement). En effet, il peut arriver que deux segments au niveau mots aient deux hypothèses de mots qui vont donner lieu à un seul segment au niveau symbole (fusion de segments de mots), mais la combinaison de deux autres hypothèses de mots des deux mêmes segments de mots vont donner lieu à deux segments de symboles indépendants. Un autre cas problématique est celui d'un segment de mot produisant deux segments symboles et d'autres hypothèses de mot du même segment ne produisent qu'un seul segment de symbole.

Dans un cas comme dans l'autre, nous avons adopté la procédure suivante : chacun des mots hypothèses d'un segment est analysé indépendamment et relativement aux mots qui sont situés dans les segments directement connectés au sien dans le graphe (cela revient à analyser tous les chemins du graphe de façon indépendante) et toutes les combinaisons de mots donnent lieu à de nouveaux segments (le mode opératoire de combinaison reste identique à celui défini précédemment en ce qui concerne les scores et positionnements temporels). Après cela, toutes les combinaisons valides de conflit entre les segments de symboles générés sont fusionnés en tenant compte des confusions entre les segments de mots initiaux.

À titre d'exemple, partons des segments de mot de la liste suivante : $\{[(quatre, 1, 0.2, 1.8, 2)]$; $[(vingts, 0.75, 1.9, 2.4, 2.9)$; $(v, 0.7, 1.9, 2, 2.5)]$; $[(un, 1, 2.3, 2.7, 3)]$ $\}$ extraite d'un graphe de mots. Il existe deux chemins valides associés à ce graphe, qui sont :
$C1 = \{[(quatre, 1, 0.2, 1.8, 2)]; [(vingts, 0.75, 1.9, 2.4, 2.9); [(un, 1, 2.3, 2.7, 3)]$ $\}$, et
$C2 = \{[(quatre, 1, 0.2, 1.8, 2)]; [(v, 0.7, 1.9, 2, 2.5)]; [(un, 1, 2.3, 2.7, 3)]$ $\}$.
Les segments en symboles associés à chacun des chemins pris indépendamment seront :
pour $C1$: $\{[(8, 0.91, 0.2, 2.1, 2.45)]; [(1, 0.91, 2.1, 2.55, 3)]$ $\}$.
pour $C2$: $\{[(4, 1, 0, 1.8, 2)], [(v, 0.7, 1.9, 2, 2.5)], [(1, 1, 2.3, 2.7, 3)];$ $\}$.

Au final, en prenant en compte les confusions inter-mots existantes initialement entre les segments en mots originels (entre *vingts* et v), on aboutit à la liste de segments en symboles totale suivante (les scores au sein d'un même segment sont normalisés pour que leur somme fasse 1) : $\{[('8', 0.48, 0.2, 1.8, 2); ('4', 0.52, 0, 1.8, 2)]$; $[('8', 0.57, 0.2, 1.8, 2); ('v', 0.43, 1.9, 2, 2.5)]; [('1', 1, 2.3, 2.7, 3)]; [('1', 0.91, 2.1, 2.55, 3)]$ $\}$.

7.5.3.2 Choix de l'association groupement/segment à retenir.

Quelque soit la façon de définir les segments audio, ils vont être combinés avec les groupements de traits (hypothèses de symboles) issus de l'écrit. À partir de là, deux problèmes majeurs sont à noter : le premier concerne la normalisation des scores issus des deux modalités. Nous avons déjà rencontré ce problème à l'occasion du chapitre précédent en ce qui concerne les expérimentations sur la fusion des deux flux au niveau des symboles isolés. Pour résoudre le problème lié à la normalisation des scores, notamment les scores acoustiques du graphe de mots de la figure 7.9, nous avons exploré les méthodes classiques utilisées dans le domaine de la fusion de données en général (de type $MinMax$, $z - score$, ...) [JNR05, WCB06]. Le second quant à lui concerne le choix des associations segment/groupement à considérer pour la suite du traitement. En d'autres termes, chaque hypothèse formulée en écrit doit être combinée avec un seul segment avant d'être envoyée au module chargé de l'interprétation de l'EM. De ce fait, lorsque plusieurs segments audio peuvent être combinés avec une même hypothèse écrit, quel couple groupement/segment retenir ?

Nous proposons trois approches pour résoudre ce problème de mise en correspondance.

A. Exploitation de l'intersection des listes des Nmeilleures étiquettes données par les deux modalités. La première méthode consiste à fusionner un segment audio et un groupement écrit en se basant sur l'analyse des étiquettes des hypothèses formulées dans chacune des deux modalités. Pour chaque segment audio, une liste de N_{audio} meilleurs labels est donnée. Pour chacun des groupements écrits également, le système de reconnaissance en charge de cette modalité associe une liste de N_{ecrit} meilleurs labels. A partir de là, un segment audio va être fusionné avec

un groupement écrit seulement si leurs listes respectives de labels ont en commun au moins une étiquette de symbole. Il en résulte qu'un segment audio peut être fusionné avec plusieurs groupements écrits. Après avoir envisagé toutes les possibilités de combinaison d'un groupement écrit avec un segment audio, il convient de ne retenir qu'une seule d'entre elles. Pour ce faire, nous avons exploré plusieurs stratégies et celle que nous avons retenue est de garder la combinaison donnant le score le plus fort pour la classe arrivant en première position. Le choix de cette méthode est validé sur la base de validation qu'on présentera plus loin.

Cette approche risque de trouver des limites dans le cas où deux symboles confusifs sont présents simultanément dans une expression. C'est le cas par exemple lorsqu'une expression contient les symboles $'2'$ et $'z'$. Il est probable alors que ces deux étiquettes soient présentes dans les listes des N_{ecrit} meilleures propositions avec des scores assez voisins. Même si la transcription de l'audio est parfaite avec la présence des mots "deux" et "z" , leur prise en compte risque ne pas aider à la désambiguïsation. En effet, chacun des deux segments audio sera valide pour être associé à chacun des deux groupements de traits.

B. Exploitation de l'intersection des listes des $Nmeilleures$ étiquettes données par les deux modalités complétée par la contrainte de positionnement relatif. Une deuxième proposition est une variante de celle décrite précédemment. En effet, en plus des étiquettes des symboles reconnus qui permettent de traiter toutes les combinaisons groupement/segment, une information additionnelle est exploitée pour décider du choix de l'association à retenir parmi toutes les hypothèses de combinaisons retenues. Il s'agit du positionnement spatial relatif du groupement écrit par rapport à la longueur (selon x :L_x^{EM}) de l'EM $P_{ecrit} = x_0^{groupement}/L_x^{EM}$ ($x_0^{groupement}$ est la position en x du début du groupement) et du positionnement temporel relatif du segment audio $P_{audio} = t_0^{segment}/T^{EM}$ par rapport à la durée totale du signal audio T^{EM} ($t_0^{segment}$ est la position en x du début du segment). L'association groupement/segment présentant le minimum de distance $|P_{ecrit} - P_{audio}|$ est considérée comme la plus optimale. Cela revient à dire qu'un segment et un groupement, pour être fusionnés, doivent satisfaire deux conditions : contenir au moins un label de symbole en commun et qu'ils soient les deux plus proches relativement à toute l'expression.

C. Utilisation d'un classifieur assurant l'alignement écrit/audio. Dans cette dernière approche, nous confions la responsabilité du choix de l'appariement entre un groupement de traits et l'ensemble des segments audio à un réseau de neurones. Celui-ci reçoit en entrée les scores des différentes classes provenant d'une part du groupement écrit ($74 + rejet$) et d'autre part les scores provenant du segment audio considéré (74). À la vue de cette double liste, Il élabore en sortie une décision pour considérer comme valide ou non l'appariement. Une fois que l'ensemble des segments audio a été balayé, seul le meilleur appariement est conservé pour être envoyé au module de fusion. Ce classifieur, tout comme celui de l'écrit (on s'en est inspiré),

est appris de façon globale. Cela revient à dire, qu'une fois une proposition de partition de l'EM complète est fait, chaque hypothèse de segmentation est analysée en considérant l'association qui lui a donnée naissance. Dans le cas de mauvaise proposition de segmentation ou de mauvaise association, cet exemple est appris comme un exemple à rejeter et en même temps, les bonnes associations impliquant les bonnes hypothèses de symboles en écrit sont apprises comme les bons exemples.

7.5.4 Fusion d'information au niveau relations

Pour la fusion au niveau des relations, ce sont les coûts de reconnaissance des relations spatiales liant les différents symboles du système écrit qui vont être modifiés. Rappelons que les coûts fournis par le système écrit aux différentes relations sont du type géométriques, ceux-ci sont très dépendants de la taille et des dispositions des symboles. Cela rend leur normalisation ardue. En tenant compte de cette contrainte pratique, nous avons exploré une fusion par approche "sac de mots" similaire à celle utilisée dans le cas des symboles. Une transformation linéaire, donnée par l'équation 7.7, est appliquée en considérant les coûts des relations alloués par la modalité manuscrite. En effet, au moment de l'analyse structurelle par le système de reconnaissance du tracé manuscrit, pour chaque couple d'hypothèses de symboles, toutes les relations les liant sont explorées et des coûts géométriques sont calculés. Chaque coût est par la suite réévalué et l'ordre des relations est de ce fait susceptible d'être changé à partir de la liste de relations possibles $lRelAudio$ issues de la modalité audio. Ceci a pour effet de pénaliser les relations absentes dans la transcription en augmentant leurs coûts. Les coûts des relations présentes en audio sont au contraire abaissés pour les favoriser lors de l'analyse grammaticale. On obtient alors :

$$Cout(R/j, lRelAudio) = \begin{cases} Coef_r \times Cout_{ecrit}(R/j) & si\ R \in lRelAudio \\ \\ Coef_p \times Cout_{ecrit}(R/j) & sinon, \end{cases} \tag{7.7}$$

où :

- $Cout_{ecrit}(R/j)$ est le coût avant fusion pour que la sous-expression j utilise la relation de type R,
- $Cout(R/j, lRelAudio)$ est le coût après fusion pour que la sous-expression j utilise la relation de type R étant donné la liste des relations données en audio $lRelAudio$,
- $Coef_r$ et $Coef_p$ sont respectivement les coefficients de rehaussement (pour les relations présentes dans $lRelAudio$: $Coef_r < 1$) et de pénalisation (pour les relations absentes dans $lRelAudio$: $Coef_p > 1$),

7.6 Résultats expérimentaux

Dans cette section nous allons présenter les divers résultats expérimentaux associés aux différentes variantes de notre système de reconnaissance d'EMs bi-modales. Nous commençons d'abord par présenter les données utilisées pour le test, ensuite les performances des systèmes mono-modaux sont données. Finalement, les performances du système complet de fusion sont rapportées et analysées.

7.6.1 Les données

Pour ce qui est du système écrit, nous avons utilisé le système que nous avons mis en œuvre pour prendre part à la compétition sur la reconnaissance des EMs manuscrites en-ligne, *CROHME 2012*. Ce système est guidé par une grammaire et un vocabulaire spécifiés par les organisateurs. À cet égard nous avons filtré la base bi-modale *HAMEX* de façon à respecter les conditions considérées en écrit. C'est ainsi que nous avons construit une sous-base $subHAMEX$ contenant une partie apprentissage pour entraîner/optimiser les divers paramètres du système de fusion des différentes méthodes exposées dans ce rapport. La base $subHAMEX$ est définie par un vocabulaire de 56 symboles différents (contre 74 pour *HAMEX*). Elle contient 1 463 EMs pour la partie apprentissage (extraites des 2 925) et 519 EMs pour le test (parmi les 1 425 disponibles en test dans *HAMEX*).

7.6.2 Performances des systèmes mono-modaux

Le système de reconnaissance des EMs manuscrites en-ligne appris sur la base d'apprentissage de *CROHME 2012*, possède les performances résumées sur le tableau 7.2. Ces expérimentations sont conduites sur l'ensemble des 519 EMs de $subHAMEX$ en considérant uniquement leur version manuscrite.

Niveau d'évaluation	Étiquettes des traits	Étiquettes des symboles	Expressions sans erreurs	Expressions à 1 erreur près	Expressions à 2 erreurs près
Taux de reco. [%]	80.05	82.93	34.10	46.44	49.52

TABLE 7.2 – Performances du système de reconnaissance des EMs manuscrites

D'après le tableau 7.2, seules 34.1% des EMs sont complètement reconnues par le système basé sur la modalité manuscrite. Toutefois, si une seule erreur en étiquette de symbole ou de relation est tolérée, on arrive déjà à 46.44% de bonne reconnaissance et à presque la moitié des EMs de bien reconnues si deux erreurs sont autorisées (49.52%). Cela suggère qu'une information extérieure pourrait faire basculer l'issue de la reconnaissance pour beaucoup d'EMs mal reconnues parce qu'une ou deux erreurs sont présentes. On peut noter que la performance du classifieur de symboles est en dessous de celle du classifieur utilisé dans le cas des symboles isolés, bien que le

nombre de classes soit plus faible (56 contre 74). En effet dans ce cas, des problèmes relatifs à la tâche de segmentation se rajoutent à ceux liés à la classification.

Le système audio en charge d'apporter l'information extérieure pour aider la reconnaissance possède les performances rapportées sur le tableau 7.3 (test sur la partie audio de l'ensemble de la base de test de *subHAMEX*).

Niveau d'évaluation	Ensemble de tous les mots	Ensemble des mots clés
Taux de reco. [%]	90.07	97.21

TABLE 7.3 – Performances du système de reconnaissance de la parole

Les performances du système de TAP sont très élevées, notamment en ce qui concerne la reconnaissance des mots clés. Ceci, combiné avec les performances du classifieur de symboles manuscrits lorsqu'il propose la bonne étiquette parmi les 5 meilleures propositions, a notamment motivé le recours à la fusion par approche "sac de mots" présentée plus haut (système audio fiable dans la majorité des cas).

Nous allons dans la suite rapporter les divers résultats relatifs au système proposé avec les différentes approches de fusion précédemment décrites dans ce rapport. La partie apprentissage de *subHAMEX* a servi pour valider les divers paramètres du système de fusion et les résultats rapportés portent sur la partie de test.

7.6.3 Performances du système basé sur la fusion par approche "sac de mots"

Les paramètres considérés dans ce cas (équations : 7.4, 7.5 et 7.6) sont validés en utilisant la partie apprentissage de la base *subHAMEX*.

Le tableau 7.4 montre les performances du système de fusion pour chacune des méthodes sur la base de test.

Méthode de fusion au niveau symboles	sans fusion au niveau symboles	fusion par filtrage (eq. 7.4)	fusion par fct linéaire (eq. 7.5)	fusion par fct sigmoïde (eq. 7.6)
Tx reco. sans fusion au niveau relations [%]	34.10	36.21	35.62	37.94
Tx de reco. avec fusion au niveau relations (eq. 7.7) [%]	36.07	37.35	39.42	40.01

TABLE 7.4 – Taux de reconnaissance des différentes configurations de fusion par approche "sac de mots" explorées et du système de référence (système de reconnaissance des EMs manuscrites)

D'après le tableau 7.4, on constate une amélioration des taux de reconnaissance globaux (au niveau expression) dans tous les cas de fusion explorés. La fusion d'information est plus bénéfique quand elle est accomplie à la fois au niveau symboles et au niveau relations (dans tous les cas de figure on récupère deux fois plus d'expressions en considérant la fusion au niveau relations). Dans le tableau 7.5 une analyse plus fine des performances du système de REM manuscrites sans fusion et avec la meilleure configuration de fusion (les deux cases en gris du tableau précédent) montre que la seconde configuration permet une amélioration des performances à tous les niveaux (étiquettes des traits, symboles, expressions si une ou deux erreurs sont autorisées).

Taux de reco.(%)	étiquettes des traits	étiquettes des symboles	expressions sans erreur	expressions à 1 erreur près	expressions à 2 erreurs près
sans fusion	80.05	82.93	34.10	46.44	49.52
avec fusion	87.31	90.09	40.01	51.58	54.71

TABLE 7.5 – Comparaison des performances du système de reconnaissance des EMs manuscrites en ligne sans fusion et avec la meilleure configuration de fusion

7.6.4 Performances du système basé sur la fusion par alignement

Ici, nous allons aborder le cas de la fusion par alignement où les informations issues de la modalité audio sont plus finement exploitées. En opérant de la sorte, on espère combler les défauts dus aux quelques erreurs commises lors de la transcription automatique de la parole. Ces erreurs peuvent être corrigées en consultant les hypothèses additionnelles à celles ayant servi à la construction de la meilleure solution.

Le tableau 7.6 donne les résultats obtenus au niveau du taux de reconnaissance parfaite des EMs complètes en considérant comme critère de choix de l'association groupement/segment uniquement l'intersection des listes des meilleurs candidats et l'association donnant le score le plus fort pour la classe en première position (Méthode A de la section 7.5.3.2).

Dans ce cas également, la fusion permet d'obtenir de meilleurs résultats comparativement au cas mono-modal basé sur l'écrit uniquement. La fusion à base de fonctions de croyance offre la meilleure amélioration dans le cadre de cette fusion (souple) et même par rapport à la meilleure des méthodes de fusion par approche "sac de mots". Ceci est en cohérence avec les divers travaux sur la pertinence de la combinaison d'information par fonctions de croyances. Ceci est grandement lié au fait que cette théorie apporte une modélisation de l'inconnue et de l'incertain. L'augmentation du nombre de poids dans le cadre de la fusion par la règle de la moyenne confirme le fait qu'il est important de ne pas tout simplement combiner de façon systématique (traiter de la même façon toutes les informations à combiner

Système évalué		Taux de reco. EMs [%]	
Système de référence : écrit seul		**34.10**	
Système de fusion : basé sur la méthode de		Trans.	Treillis
Moyenne arithmétique	**simple**	**37.18**	**37**
	pondérée par les taux de reco. globaux	**36.71**	**36.41**
	pondérée par les taux de reco. par classe	**37.87**	**37.87**
Règle du produit		**35.04**	**34.87**
Règle du maximum		**35.52**	**35.26**
Règle de Borda		**35.83**	**36.42**
Fonctions de croyance		**41.43**	**41.82**

TABLE 7.6 – Taux de reconnaissance des différentes configurations de fusion par alignement explorées et du système de référence (modalité manuscrite) en considérant la meilleure solution audio uniquement (Trans.) et le treillis de mots (Treillis) en se basant que sur l'intersection des listes de label pour le choix des associations groupement/segment (Méthode A de la section 7.5.3.2).

arrivant à l'unité de fusion) sans des pondérations adéquates. La méthode de Borda qui s'est avérée très pertinente lors de la fusion des symboles isolés, ne l'est plus dans ce contexte. Ceci est principalement dû au fait que les propositions du système de TAP ne sont pas suffisamment diversifiées et ne permettent pas de remédier aux cas des transcriptions incorrectes.

Sur le tableau 7.7 est rapportée la comparaison du système basé sur la meilleure configuration de fusion à base de fonctions de croyance et le système mono-modal basé sur le signal manuscrit en-ligne uniquement.

Dans ce scénario de fusion également, l'amélioration est ressentie aussi bien au niveau expression qu'aux niveaux inférieurs (traits et symboles). Les taux d'expression présentant seulement une seule ou deux erreurs sont également augmentés, cela veut dire qu'il est encore possible d'améliorer les taux globaux en analysant les expressions présentant ce type de problèmes.

Dans le tableau 7.8, on donne les résultats obtenus au niveau du taux de reconnaissance parfaite des EMs complètes en considérant cette fois-ci à la fois le critère d'intersection entre les listes des $N - meilleurs$ candidats mais aussi la proximité relative des groupements et segments combinés (Méthode B, *cf.* section 7.5.3.2).

Ici aussi, la fusion permet d'obtenir de meilleurs résultats comparativement au cas mono-modal. L'ajout de la contrainte de proximité des groupements et segments

Taux de reco.(%)	étiquettes des traits	étiquettes des symboles	expressions sans erreur	expressions à 1 erreur près	expressions à 2 erreurs près
sans fusion	80.05	82.93	34.10	46.44	49.52
avec fusion	86.73	88.21	41.82	50.67	53.37

TABLE 7.7 – Comparaison des performances du système de reconnaissance des EMs manuscrites en ligne sans fusion et avec la meilleure configuration de fusion à base de fonction de croyances et en considérant un alignement par l'intersection des listes des N-meilleurs candidats (Méthode *A*, *cf.* section 7.5.3.2).

Système évalué		**Taux de reco. EMs [%]**	
Système de référence : écrit seul		**34.10**	
Système de fusion : basé sur la méthode de		Trans.	Treillis
Moyenne arithmétique	**simple**	**37.38**	**37.19**
	pondérée par les taux de reco. globaux	**36.9**	**36.6**
	pondérée par les taux de reco. par classe	**37.87**	**38.06**
Règle du produit		**35.23**	**35.06**
Règle du maximum		**35.71**	**35.45**
Règle de Borda		**36.02**	**36.61**
Fonctions de croyance		**41.61**	**42.02**

TABLE 7.8 – Taux de reconnaissance des différentes configurations de fusion par alignement explorées et du système de référence (modalité manuscrite) en considérant la meilleure solution audio uniquement (Trans.) et le treillis de mots (Treillis) en se basant sur l'intersection des listes de label des meilleurs candidats ainsi que sur les positions relatives des segments et groupements pour le choix des associations groupement/segment (Méthode *B*, *cf.* section 7.5.3.2).

combinés permet de rehausser légèrement les résultats de la fusion. Néanmoins, cette nouvelle considération ne fait pas évoluer les performances des méthodes les unes par rapport aux autres. En effet, la fusion à base de fonctions de croyance offre, cette fois-ci encore, la meilleure amélioration dans le cadre de cette fusion. Ceci est certainement lié au fait que toutes ces méthodes sont toutes sensibles à l'alignement des signaux des deux modalités.

Cette nouvelle condition de choix de l'association à considérer pour la fusion

permet, de la même manière que pour les méthodes précédentes, d'apporter une amélioration aux niveaux traits et symboles. En tolérant une ou deux erreurs d'étiquette pour les EMs, les taux sont également augmentés. Le tableau 7.9 illustre cette situation en rapportant la comparaison du système basé sur la meilleure configuration de fusion à base de fonctions de croyance et le système mono-modal de référence.

Taux de reco.(%)	étiquettes des traits	étiquettes des symboles	exp. sans erreur	expressions à 1 erreur près	expressions à 2 erreurs près
sans fusion	80.05	82.93	34.10	46.44	49.52
avec fusion	86.68	88.43	42.02	50.67	53.18

TABLE 7.9 – Comparaison des performances du système de reconnaissance des EMs manuscrites en ligne sans fusion et avec la meilleure configuration de fusion à base de fonction de croyances en considérant, en plus de l'intersection des listes des N-meilleurs candidats, la contrainte de proximité en position relative (Méthode B, *cf.* section 7.5.3.2).

La comparaison des tableaux 7.5, 7.7 et 7.9 montre que pour les meilleures configurations de fusion (fusion par approche "sac de mots" à base de sigmoïde et fusion par alignement à base de fonctions de croyances) la fusion à base de fonction de croyances permet de récupérer plus de cas mal reconnus initialement (encore plus quand l'alignement est basé sur les positions relatives). Cela se répercute sur les taux de reconnaissance autorisant une ou deux erreur(s) qui sont plus bas que dans le cas de la fusion par approche "sac de mots" (plus d'exemples initialement ayant une ou deux erreurs sont devenus sans erreur).

En ce qui concerne l'utilisation de la dernière approche d'alignement écrit/audio basée sur de la classification, il est nécessaire de disposer de la vérité terrain au niveau alignement des deux modalités. En effet, dans la mesure où l'apprentissage global du classifieur a pour vocation de d'apprendre à rejeter deux types d'exemples :

- Les exemples dont l'hypothèse de segmentation formulée à l'écrit est mauvaise. Nous remédions à ce cas en considérant le classifieur écrit déjà entraîné ce qui aura pour effet de rendre disponible l'information de rejet de l'hypothèse écrit sur la classe de rejet du classifieur écrit (selon la qualité de l'apprentissage écrit). Le classifieur de fusion aura dans ce cas à copier ce rejet formulé en écrit dans sa propre classe de rejet.
- Les exemples issus d'un mauvais alignement écrit/audio, abstraction faite de la qualité de l'hypothèse de segmentation de l'écrit et de la reconnaissance qui lui est associée. Ce type d'exemple requiert la disponibilité de la vérité terrain sur les associations groupement/segment. Ceci peut être problématique dès lors qu'une classe de symbole se retrouve plusieurs fois dans la même EM.

Dans la mesure où notre base n'est pas annotée à ce niveau, nous présentons ici,

une expérimentation préliminaire qui ne considère que les EMs qui ne contiennent pas la même classe de symbole à plusieurs reprises (une classe est présente une seule fois au maximum dans l'EM). Cette considération permet de générer automatiquement l'alignement vérité terrain.

Les données utilisées pour cette expérimentation sont extraites de $subHAMEX$ en ne gardant que celles respectant la contrainte de présence une fois au maximum de chaque classe de symboles. Elles sont au nombre de 731 EMs pour l'apprentissage (549 pour l'entraînement et 182 pour la validation des poids optimaux) et 239 pour le test. Cette nouvelle base de test est du fait de cette contrainte moins complexe. En effet, même si toutes les relations spatiales et tous les symboles pilotant notre système sont présents, la taille moyenne en symbole des EMs est plus faible relativement aux EMs de $subHAMEX$ (5 symboles en moyenne, avec la plus courte écrite avec 3 symboles et la plus longue avec 12 symboles contre 13 pour $subHAMEX$ avec 3 symboles pour la plus courte et 28 pour la plus longue).

Le classifieur utilisé ici est un simple PMC avec une seule couche cachée, ayant $2 * Nbclasses + 1 = 113$ neurones sur la couche d'entrée, 100 neurones sur la couche cachée et 57 neurones sur la couche de sortie.

Sur le tableau 7.10 sont rapportés les résultats de cette étude préliminaire. Les performances du système de référence et de la meilleure configuration de fusion obtenue précédemment par les fonctions de croyance sont également représentées (évalués sur la nouvelle sous-base contenant les 239 EMs qui ne contiennent chacune chaque étiquette de symboles une fois au maximum).

Taux de reco.(%)	étiquettes des traits	étiquettes des symboles	expressions sans erreur	expresssions à 1 erreur près	expressions à 2 erreurs près
Écrit seul	86.16	88.93	41.84	74.89	76.15
Fusion par les fonctions de croyance	90.95	93.06	59.83	77.4	77.8
Fusion par classification	92.18	92.49	62.76	79.9	79.9

TABLE 7.10 – Comparaison des performances du système de reconnaissance des EMs manuscrites en ligne sans fusion et avec la meilleure configuration de fusion à base de fonction de croyances ainsi que la fusion à base du classifieur PMC (Méthode C, *cf.* section 7.5.3.2)

On peut observer sur la table 7.10 que, même si le problème est simplifié en ne prenant que des EMs ayant une seule fois au maximum chaque étiquette de symbole, la méthode de fusion par classification est celle qui assure la meilleure performance au niveau EM complète (que l'on autorise ou pas d'erreurs). On note également que la fusion à base de fonctions de croyance donne dans ce cas un gain nettement

plus important, par rapport à la modalité manuscrite, que lors des précédentes expérimentations avec une base de test contenant des EMs ayant plusieurs fois la même étiquette. Cette étude préliminaire sur l'alignement des signaux écrit et sonore laisse penser que la source principale d'ambiguïté qui subsiste dans notre système bi-modal est liée à la confusion existante lorsque la même étiquette est présente plusieurs fois, à différents endroits dans l'EM.

7.7 Quelques exemples réels de reconnaissances

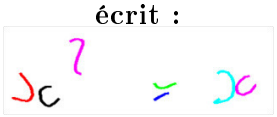
Dans le tableau 7.11 sont rapportés quelques exemples illustrant l'approche du traitement bi-modal à la reconnaissance des EMs. L'analyse du déroulement du processus de fusion y est également faite.

Tracé manuscrit et transcription automatique	Résultats de reconnaissance basés sur l'écrit seul	Résultats de reconnaissance après fusion
Gain dû à la fusion au niveau des relations uniquement		
écrit :  **audio :** "j indice un égale un" **vérité terrain :** $j_1 = 1$	$11 = 1$ coûts des relations en cause : $C_{Subscript} = 1681.34$, $C_{Hpaire} = 672.83$	$j_1 = 1$ coûts des relations en cause : $C_{Subscript} = 672.54$, $C_{Hpaire} = 1412,94$
Grâce à la fusion au niveau relation uniquement ($Coef_r = 0.4$ et $Coef_p = 2.1$, fixés expérimentalement sur la base de validation de fusion), l'ordre des relations paire horizontale ($Hpaire$) et indice ($Subscript$), liant les deux premières hypothèses de symboles (les plus à gauche), est changé. Ce changement fait que le choix de l'étiquette de la première hypothèse bascule de 1 à j au moment de l'interprétation (exploitation de la liste des Nmeilleures étiquettes associées à une hypothèse de symbole pour satisfaire la grammaire pilotant la reconnaissance).		

Ascendant de la fusion par transformation sigmoïdale par rapport aux autres méthodes de fusion par approche "sac de mots"

écrit :  **audio :** "a entre parenthèses x" **vérité terrain :** $k(x)$	$k1x^1$ scores des hypothèses en causes ($h_{\{0\}}$ et $h_{\{0,1\}}$) : $S(k/h_{\{0\}}) = 0.87$, $S(A/h_{\{0,1\}}) = 0.4$	**seuillage ou transformation linéaire :** Ax^1 **transformation sigmoïdale :** $k(x)$ Score après fusion par sigmoïde : $S(k/h_{\{0\}}) = 0.85$, $S(A/h_{\{0,1\}}) = 0.54$

Les méthodes de fusion par approche "sac de mots" basées sur le seuillage ou la transformation linéaire ne permettent pas de bien reconnaître l'EM et rendent l'interprétation encore plus lointaine de la vraie solution (on trouve comme résultat 'Ax^1'). Ceci est dû au fait que dans ce cas, l'information apportée par l'audio est erronée concernant le premier symbole (substitution du 'k' par un 'a'), ce qui est un inconvénient majeur pour les deux approches citées plus haut. La méthode basée sur la transformation non linéaire, assure une meilleure prise en compte à la fois de l'information issue de l'audio et de la confiance du classifieur écrit. Ceci permet, en dépit de l'erreur de la proposition du système audio, de faire valoir le fort score alloué à la bonne hypothèse de segmentation proposé en écrit.

<table>
<tr><td colspan="3">Fusion par fonctions de croyances</td></tr>
<tr><td>écrit :
 
audio :
“s au carré égale x”

vérité terrain :
$x^2 = x$</td><td>$jc^2 = x$

Deux meilleures étiquettes de symboles pour l’hypothèse $h_{\{0,1\}}$:
$S(rejet/h_{\{0,1\}}) = 0.84$,
$S(x/h_{\{0,1\}}) = 0.15$</td><td>$x^2 = x$

Après fusion :
$S(x/h_{\{0,1\}}) = 0.14$,
$S(rejet/h_{\{0,1\}}) = 0.059$</td></tr>
<tr><td colspan="3">Pour cet exemple, la fusion à base de fonctions de croyance est la seule méthode qui permet que celui-ci soit reconnu après fusion. En effet, les deux premiers traits (rouge et noir) étant écrits relativement loin l’un de l’autre, le classifieur écrit suggère qu’ils ne soient pas associés à une même hypothèse en proposant un score de rejet élevé 0.84. Toutefois, la seconde hypothèse d’étiquette est la bonne classe (‘x’) même si ce n’est qu’avec un très faible score comparativement au rejet (0.15). Ces deux traits pris indépendamment ont des formes respectives en écrit qui soit proches d’autres classes (‘j’ pour le premier et ‘c’ pour le second). C’est cela, qui conduit au résultat de reconnaissance incluant ces deux classes lors de l’analyse du signal écrit seul. Dans le signal audio, le premier segment est étiqueté ‘s’avec un score de 0.48 et ‘x’ avec un score de 0.45. L’étiquette ‘x’ est commune aux deux modalités et permet donc de fusionner les groupement écrit et le segment audio. Le fort score de la classe du rejet au niveau écrit met en échec toutes les méthodes à base de règle à l’exception de la méthode de fusion par fonctions de croyance qui permet d’attribuer à l’hypothèse en cours un score suffisamment haut pour l’étiquette ‘x’ et la replacer en meilleure proposition. À noter que les scores après fusion donnés ici ne sont pas renormalisés (ils correspondent aux masses calculées après fusion).</td></tr>
</table>

Table 7.11 – Commentaires sur quelques exemples réels

7.8 Bilan

Dans ces expérimentations, nous avons abordé le problème de la reconnaissance d'expressions mathématiques complètes dans le cadre d'un traitement bi-modal. Les modalités audio et écriture manuscrite en-ligne se sont avérées être effectivement complémentaires. Les résultats obtenus vont dans le sens des conclusions que nous avions énoncées à l'issue du chapitre précédent concernant la fusion au niveau de la reconnaissance des symboles uniquement. On arrive à améliorer les performances du système global de 21.5% (avec la fusion à base de fonctions de croyances). Un autre point important est lié à l'apport de la fusion aux deux niveaux (symboles et relations) dans la mesure où la modalité audio aide à l'identification des symboles et des relations à la fois. En effet, même si les relations identifiées en audio restent sans contexte dans notre cas (les symboles impliqués dans cette relation ne sont pas considérés), elles constituent tout de même un filtre qui peut être pertinent au moment de l'analyse structurelle opérée sur le tracé manuscrit.

L'étude préliminaire rapportée dans la dernière section suggère que l'utilisation d'une fusion à base de classifieurs peut avoir un double apport. D'une part, ces méthodes ont le meilleur comportement face au problème de normalisation (à l'image de ce qui a été rapporté au chapitre précédent sur les symboles isolés). D'autre part, le recours à une méthode d'apprentissage globale de ces classifieurs a pour effet de lui procurer un pouvoir de rejet des mauvais alignements audio/écrit. Néanmoins, le problème de l'alignement des symboles audio/écrits sur l'ensemble des EMs sans tenir compte des occurrences de chaque étiquette de symbole reste à résoudre.

Conclusion générale

Conclusions

Le travail de thèse rapporté dans ce manuscrit s'inscrit dans le cadre du projet région DEPART (Documents Ecrits et Paroles - Reconnaissance et Traduction). Ce projet vise à constituer au niveau de la région des Pays de la Loire un pôle de compétences unique en France associant analyse du signal audio et manuscrit au traitement automatique des langues. Il a pour but principal notamment de résoudre des problèmes scientifiques et technologiques particulièrement difficiles en mettant en jeu des données multi-modales.

Les langages graphiques, comme les mathématiques, font partie des problèmes qui rentrent dans cette catégorie et qui requièrent ce type d'approches multi-modales. L'objet de la présente thèse était donc d'apporter des solutions quant à l'exploitation conjointe des flux écrit et sonore afin d'améliorer la reconnaissance automatique des expressions mathématiques.

Avant de s'intéresser au problème de la reconnaissance en soi, l'originalité de la problématique traitée a imposé la construction d'une base de données ayant pour objet de proposer chacune des expressions mathématiques sous les deux formats impliqués dans ce travail (le tracé manuscrit en-ligne de l'EM et l'enregistrement de sa dictée). C'est ainsi que nous avons collecté et complètement annoté cette base originale et unique, que nous avons nommée : *HAMEX*.

Une fois les données à disposition, dans un premier temps, afin de valider les hypothèses de complémentarité des signaux audio et écrit, nous avons exploré une première expérimentation en traitant le problème simplifié des symboles isolés. Nous avons pour cela, après une étude bibliographique, fixé le cadre dans lequel est construit le système bi-modal proposé pour la reconnaissance des symboles isolés. Afin de combiner les informations audio et écrit de façon efficace et pertinente, nous avons opté pour une fusion tardive, et nous avons utilisé différentes méthodes classiques de combinaison d'informations. Cette étude a permis de valider définitivement cette complémentarité pressentie des deux modalités. De plus, elle a permis de dégager, parmi la panoplie des méthodes explorées, les plus appropriées dans le cas d'un traitement multi-modal.

Après cette première étude exploratoire, nous nous sommes attaqué au problème complet en s'intéressant à l'ensemble des étapes conduisant à l'interprétation d'une EM. La consultation de la littérature a mis en valeur un ensemble de constatations :

- Les systèmes d'interprétation s'appuyant séparément sur l'une ou l'autre des deux modalités en cause dans ce travail (écrit et audio) n'ont pas la même maturité.
- La représentation des EMs à l'écrit est spatiale et se prête plus facilement à l'interprétation des EMs qu'en audio, où les informations sur les symboles et surtout les relations entre les symboles ne sont pas toujours clairement identifiables.
- Il existe énormément de systèmes dédiés à la tâche d'interprétation des EMs manuscrites en-ligne seules contrairement au cas de la modalité de la parole, où les rares travaux préconisent de disposer de modes d'interaction supplé-

mentaires pour arriver à l'interprétation finale.

Nous avons donc proposé un système bi-modal, de type modulaire, dans lequel le module chargé du traitement du tracé manuscrit, est le module principal. C'est au niveau de ce module que sont accomplies les diverses étapes d'interprétation. Le second module se charge de la modalité audio pour fournir une information supplémentaire pour guider la reconnaissance. Cela se fait en remettant en cause les propositions faites par la partie en charge du signal manuscrit et en proposant éventuellement des corrections (suivant la méthode de fusion), à travers les différentes unités du module de fusion. La modularité du système proposé, permet de procéder assez facilement à différentes stratégies de fusion en exploitant différentes méthodes.

Nous avons proposé deux principales méthodologies de fusion, une à base de fusion par approche "sac de mots" où on considère que le système audio est complètement fiable. Dans ce cas, c'est ce système qui décide du sort des hypothèses formulées par le module de l'écrit. Une deuxième méthodologie, plus souple, consiste à prendre en compte de façon équilibrée les propositions de chacune des deux modalités et en les alignant. Dans un cas comme dans l'autre nous avons exploité plusieurs approches de fusion.

Toutes les propositions faites dans cette thèse pour le traitement bi-modal des EMs, ont montré une amélioration sensible des performances globales (au niveau de leur interprétation complète), mais aussi aux niveaux les plus bas. En effet, les propositions de symboles formulées par le système écrit sont meilleures dans le cadre bi-modal (où les décisions sont prises en consultant conjointement les flux audio et écrit), signe que la modalité audio aide dans plusieurs cas (souvent quand les classes en conflit impliquent la bonne classe et que les scores sont très proches) à prendre la meilleure solution et à pénaliser les mauvaises hypothèses de symboles.

Perspectives

Il existe des retombées immédiates de ces travaux qui peuvent faire naturellement l'objet d'études plus approfondies (stages de recherches ou thèse). En effet, après ces travaux de thèse quelques pistes émergent naturellement comme une (des) suite(s) logique(s).

- L'exploitation des techniques de classification, qui ont fait leur preuve dans le cadre de la fusion au niveau de la reconnaissance des symboles isolés pourrait avoir un impact majeur dans l'amélioration du système actuel proposé. En effet, dans la dernière approche d'alignement que nous avons considérée, le classifieur est en charge de proposer le meilleur alignement groupement écrit/-segment audio. Une fois cet alignement proposé, toutes les méthodes de fusion présentées dans ce manuscrit peuvent être utilisées pour obtenir le score issu de la combinaison des deux modalités. L'objectif, une fois toute la base annotée au niveau alignement écrit/audio pour avoir suffisamment de données en volume et en diversité, serait d'exploiter directement les scores issus du classifieur lui-même.
- Un autre point concerne l'élargissement de l'espace de recherche de la meilleure interprétation. En effet, dans la configuration actuelle, lors de l'application des règles de la grammaire, le graphe sur lequel celles-ci sont appliquées est formé de sorte que chaque hypothèse de l'écrit ne soit présente qu'une fois. Cela veut dire qu'à l'issu de la fusion, une hypothèse issue de l'écrit est combinée avec une seule hypothèse audio qui est présumée être la meilleure association au sens des méthodes d'alignement proposées ici (méthodes A, B et C de la section 7.5.3.2) et on ne laisse pas la possibilité d'en garder plusieurs pour permettre à l'étape d'interprétation de choisir la meilleure. Pour faire cela, il est nécessaire de mettre en place une grammaire $2D$ bi-modale capable de remettre en cause non seulement les groupements de traits de l'écrit retenus mais aussi de prendre en compte la validité de l'alignement des signaux audio et écrit.
- Le dernier point est lié à l'exploitation du contexte local des symboles/relations. Ceci en introduisant la notion de "modèle de langage" au moment de la construction des alignement écrit/audio. Il ne serait plus question de fusionner les symboles de façon isolés mais des sous-expressions formées d'un certain nombre d'hypothèses de symboles. Ceci pourrait augmenter la précision des alignements audio/écrit et donc permettrait de gagner en robustesse.

Bibliographie personnelle

- Multimodal Mathematical Expressions Recognition : case of speech and handwriting, **Medjkoune, S.** Mouchère, H. Petitrenaud, S. Viard-Gaudin, HCI International, Human-Computer Interaction. Interaction Modalities and Techniques. Dans Lecture Notes in Computer Science, 2013, Las Vegas, Nevada, USA, vol. 8007, pp. 77-86, Springer Berlin Heidelberg.

- Using online handwriting and audio streams for mathematical expressions recognition : a bimodal approach. **Medjkoune, S.** Mouchère, H. Petitrenaud, S. Viard-Gaudin, C. Document Recognition and Retrieval XX, Part of the IS&T/SPIE 27th Annual Symposium on Electronic Imaging, 2013, San Francisco, CA USA, pp. 865810–865810.

- Fusion d'Informations Bi-modales pour la Reconnaissance d'Expressions Mathématiques Cas des modalités audio et écriture manuscrite en ligne. **Medjkoune, S.** Mouchère, H. Petitrenaud, S. Viard-Gaudin, C. Colloque International Francophone sur l'Ecrit et le Document (CIFED), 2012, Bordeaux, France.

- Using Speech for Handwritten Mathematical Expression Recognition Disambiguation. **Medjkoune, S.** Mouchère, H. Petitrenaud, S. Viard-Gaudin. International Conference on Frontiers in Handwriting Recognition (ICFHR), 2012, Bari, Italie, pp. 1-6.

- HAMEX - a Handwritten and Audio Dataset of Mathematical Expressions. Quiniou, S. Mouchère, H. Peã Saldarriaga, S. Viard-Gaudin, Ch. Morin, E. Petitrenaud, S. **Medjkoune, S.** 11th International Conference on Document Analysis and Recognition (ICDAR), 2011, Beijing, Chine, pp. 452–456.

- Handwritten and Audio Information Fusion for Mathematical Symbol Recognition. **Medjkoune, S.** Mouchère, H. Petitrenaud, S. Viard-Gaudin, C. 11th International Conference on Document Analysis and Recognition (ICDAR), 2011, Beijing, Chine, pp. 379–383.

- **Medjkoune, S.**, Handwritten and Audio Information Fusion for Mathematical Symbol Recognition, First Sino-French Workshop on Education and Research collaborations in Information and Communication Technologies (SIFWICT), 2011, Nantes, France.

Bibliographie

[ADHB+04] Martine Adda-Decker, Benoît Habert, Claude Barras, Gilles Adda, P Boula de Mareüil, and Patrick Paroubek. Une étude des disfluences pour la transcription automatique de la parole spontanée et l'amelioration des modèles de langage. *Actes des 25èmes Journees d'Études sur la Parole (JEP)*, 2004. (Cité en page 109.)

[AFMVG11] Ahmad-Montaser Awal, Guihuan Feng, Harold Mouchere, and Christian Viard-Gaudin. First experiments on a new online handwritten flowchart database. In *IS&T/SPIE Electronic Imaging*, pages 78740A–78740A. International Society for Optics and Photonics, 2011. (Cité en page 19.)

[AH71] Bishnu S Atal and Suzanne L Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, 50 :637, 1971. (Cité en page 47.)

[AH78] Nijenhuis Albert and S. Wilf Herbert. *Combinatorial algorithms for computers and calculators*, chapter 28-29, pages 267–282. New York : Academic Press, 2nd edition, 1978. (Cité en page 80.)

[AHESK10] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. Multimodal fusion for multimedia analysis : a survey. *Multimedia systems*, 16(6) :345–379, 2010. (Cité en pages 58, 60, 61, 62, 63 et 65.)

[AK10] MA Anusuya and SK Katti. Speech recognition by machine, a review. *International Journal of Computer Science and Information Security*, 6(3) :181–205, 2010. (Cité en page 45.)

[AMVG09] Ahmad-Montaser Awal, Harold Mouchère, and Christian Viard-Gaudin. Towards handwritten mathematical expression recognition. In *Tenth International Conference on Document Analysis and Recognition*, pages 1046–1050, 2009. (Cité en page 37.)

[AMVG10] Ahmad-Montaser Awal, Harold Mouchère, and Christian Viard-Gaudin. A hybrid classifier for handwritten mathematical expression recognition. In *IS&T/SPIE Electronic Imaging*, pages 753410–753410. International Society for Optics and Photonics, 2010. (Cité en page 136.)

[AMVG12] Ahmad-Montaser Awal, Harold Mouchère, and Christian Viard-Gaudin. A global learning approach for an online handwritten mathematical expression recognition system. *Pattern Recognition Letters*, 2012. In Press, Corrected Proof. (Cité en pages 30, 33, 36, 37, 38, 39, 97, 98 et 102.)

[And67] Robert H. Anderson. Syntax-directed recognition of hand-printed two-dimensional mathematics. In *Symposium on Interactive Systems for Experimental Applied Mathematics : Proceedings of the Association for Computing Machinery Inc. Symposium*, pages 436–459, New York, NY, USA, 1967. ACM. (Cité en pages 24 et 35.)

[ÁSB12] Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models. *Pattern Recognition Letters*, 2012. (Cité en pages 34, 36 et 38.)

[Awa10] Ahmad-Montaser Awal. *Reconnaissance de structures bidimensionnelles : Application aux expressions mathématiques manuscrites en-ligne*. PhD thesis, Université de Nantes, 2010. (Cité en pages 33, 34, 36, 39, 94, 100, 101, 102, 103, 104, 106, 134, 136, 137, 138 et 205.)

[AY92] Belaid Abdel and Belaid Yolande. Reconnaissance des formes : Methodes et applications, 1992. (Cité en page 32.)

[BAB+07] Henrik Boström, Sten F Andler, Marcus Brohede, Ronnie Johansson, Alexander Karlsson, Joeri van Laere, Lars Niklasson, Marie Nilsson, Anne Persson, and Tom Ziemke. On the definition of information fusion as a field of research. 2007. (Cité en page 58.)

[Bai88] H.S. Baird. Applications of multidimensional search to structural feature identification. In Gabriel Ferraté, Theo Pavlidis, Alberto Sanfeliu, and Horst Bunke, editors, *Syntactic and Structural Pattern Recognition*, volume 45 of *NATO ASI Series*, pages 137–149. Springer Berlin Heidelberg, 1988. (Cité en page 32.)

[BBNN93] E. J. Bellegarda, J. R. Bellegarda, D Nahamoo, and K.S. Nathan. A probabilistic framework for on-line handwriting recognition. In *Third International Workshop Frontiers of Handwriting Recognition*, pages 225–234, 1993. (Cité en page 34.)

[BC07] H. Bredin and G. Chollet. Audio-visual speech synchrony measure for talking-face identity verification. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–233–II–236, 2007. (Cité en page 72.)

[BD04] Niels Ole Bernsen and Laila Dybkjær. Evaluation of spoken multimodal conversation. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 38–45. ACM, 2004. (Cité en page 66.)

[BDMD+07] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Theodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al. Automatic speech recognition and speech variability : A review. *Speech Communication*, 49(10) :763–786, 2007. (Cité en page 45.)

[BG97] Dorothea Blostein and Ann Grbavec. *Recognition of mathematical notation*, chapter 22, pages 557–582. World Scientific Publishing Company, 1997. (Cité en page 24.)

[BH84] A. Belaïd and J. P. Haton. A syntactic approach for hand-written mathematical formula recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984. (Cité en pages 32 et 35.)

[BHA+01] I. Bloch, A. Hunter, A. Ayoun, S. Benferhat, P. Besnard, L. Cholvy, R. Cooke, D. Dubois, and H. Fargier. Fusion : general concepts and characteristics. *International Journal of Intelligent Systems*, 16 :1107–1134, 2001. (Cité en page 58.)

[BHB02] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. Online handwriting recognition with support vector machines-a kernel approach. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 49–54. IEEE, 2002. (Cité en page 34.)

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. (Cité en pages 33, 34, 197 et 199.)

[BJB+08] Thierry Bazillon, Vincent Jousse, Frédéric Béchet, Yannick Estève, Georges Linarès, and Daniel Luzzati. La parole spontanée : transcription et traitement. *Revue TAL. Volume*, 49(3), 2008. (Cité en page 109.)

[BK88] Glenn Baptista and KM Kulkarni. A high accuracy algorithm for recognition of handwritten numerals. *Pattern Recognition*, 21(4) :287–291, 1988. (Cité en page 32.)

[BKB10] Anjali Bala, Abhijeet Kumar, and Nidhika Birla. Voice command recognition system based on mfcc and dtw. *International Journal of Engineering Science and Technology*, 2(12) :7335–7342, 2010. (Cité en pages 112 et 113.)

[Blo93] Isabelle Bloch. *fusion d'informations en traitement du signal et des images*. Traité IC2, série Traitement du signal et de l'image. Lavoisier, 1993. (Cité en pages 58 et 64.)

[Blo96] I. Bloch. Information combination operators for data fusion : a comparative review with classification. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 26(1) :52–67, 1996. (Cité en page 61.)

[BNPW07] Regina Bernhaupt, David Navarre, Philippe Palanque, and Marco Winckler. Model-based evaluation : A new way to support usability evaluation of multimodal interactive applications. *Maturing Usability : Quality in Software, Interaction and Quality, series on HCI. Springer (April 2007)*, 2007. (Cité en page 66.)

[Bol79] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(2) :113–120, 1979. (Cité en page 111.)

[BPWN07] Regina Bernhaupt, Philippe Palanque, Marco Winckler, and David Navarre. Usability study of multi-modal interfaces using eye-tracking. In *Human-Computer Interaction–INTERACT 2007*, pages 412–424. Springer, 2007. (Cité en page 66.)

[Bru95] Jacob Bruno. *Un outil informatique de gestion de Modèles de Markov Cachés : expérimentations en Reconnaissance Automatique de la Parole*. PhD thesis, Université de Paul Sabatier de Toulouse III, 1995. (Cité en pages 46 et 47.)

[Bun97] Laurent Buniet. *Traitement automatique de la parole en milieu bruité : étude de modèles connexionnistes statiques et dynamiques*. PhD thesis, Université Henri Poincaré - Nancy 1, 1997. (Cité en page 204.)

[Caj28] Florian Cajori. *A history of mathematical notations*, volume 1. Courier Dover Publications, 1928. (Cité en pages 10 et 11.)

[CBN04] Laurence Cholvy, Jérôme Besombes, and Vincent Nimier. Information evaluation in fusion : a case study. In *10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004)*, pages 993–1000. Citeseer, 2004. (Cité en page 66.)

[CBSV03] Kyong Chang, Kevin W Bowyer, Sudeep Sarkar, and Barnabas Victor. Comparison and combination of ear and face images in appearance-based biometrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9) :1160–1165, 2003. (Cité en page 68.)

[CD00] Ross Cutler and Larry Davis. Look who's talking : speaker detection using video and audio correlation. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 3, pages 1589–1592 vol.3, 2000. (Cité en page 68.)

[CER+07] Arthur Chan, Gouvea Evandro, Singh Rita, Ravishankar Mosur, Rosenfeld Ronald, Sun Yitao, Huggins-Daines David, and Seltzer Mike. *The Hieroglyphs : Building Speech Applications Using CMU Sphinx and Related Resources*. http ://speech.tifr.res.in/tutorials/sphinxDocChan070111.pdf, 2007. (Cité en pages 45, 46, 48, 49, 50, 51 et 139.)

[CG96] Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996. (Cité en page 50.)

[Cho89] Philip A Chou. Recognition of equations using a two-dimensional stochastic context-free grammar. In *1989 Advances in Intelligent Robotics*

Systems Conference, pages 852–865. International Society for Optics and Photonics, 1989. (Cité en page 36.)

[CJM+97] Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. Quickset : multimodal interaction for distributed applications. In *Proceedings of the fifth ACM international conference on Multimedia*, MULTIMEDIA '97, pages 31–40, New York, NY, USA, 1997. ACM. (Cité en page 66.)

[CM02] Antoine Cornuéjols and Laurent Miclet. *Apprentissage artificiel : concepts et algorithmes*. Eyrolles, 2002. (Cité en pages 33, 114, 197, 199 et 201.)

[Cou96] Bertrand Couasnon. *Segmentation et reconnaissance de documents guidées par la connaissance a priori : application aux partitions musicales*. PhD thesis, Université de Rennes I, France, 1996. (Cité en page 30.)

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3) :273–297, 1995. (Cité en pages 34 et 121.)

[CWA83] Larry A Chang, CM White, and L Abrahamson. Handbook for spoken mathematics. *Lawrence Livermore National Laboratory*, 1983. (Cité en page 12.)

[CY00] Kam-Fai Chan and Dit-Yan Yeung. Mathematical expression recognition : a survey. *International Journal on Document Analysis and Recognition*, 3(1) :3–15, 2000. (Cité en pages 24 et 32.)

[CY01] Kam-Fai Chan and Dit-Yan Yeung. Pencalc : A novel application of on-line mathematical expression recognition technology. In *In Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 774–778, 2001. (Cité en page 29.)

[dB81] Jean-Charles de Borda. Mémoire sur les élections au scrutin. In *Mémoires de l'Académie Royale des Sciences*, pages 657–665. 1781. (Cité en pages 61, 62 et 119.)

[DBB52] KH Davis, R Biddulph, and S Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24 :637, 1952. (Cité en pages 45 et 47.)

[DE64] M. R. Davis and T. O. Ellis. The rand tablet : a man-machine graphical communication device. In *Proceedings of the October 27-29, 1964, fall joint computer conference, part I*, AFIPS '64 (Fall, part I), pages 325–331, New York, NY, USA, 1964. ACM. (Cité en page 24.)

[Dem67] Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, 38(2) :325–339, 1967. (Cité en page 63.)

[Dem68] A. P. Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(2) :pp. 205–247, 1968. (Cité en page 57.)

[DHS01] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2001. (Cité en pages 33, 34, 114, 197, 199 et 201.)

[DHT00] Christian Heide Damm, Klaus Marius Hansen, and Michael Thomsen. Tool support for cooperative object-oriented design : gesture based modelling on an electronic whiteboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 518–525. ACM, 2000. (Cité en page 30.)

[DKZ+03] P.M. Djurić, Jayesh H. Kotecha, Jianqui Zhang, Yufei Huang, T. Ghirmai, M.F. Bugallo, and Joaquin Miguez. Particle filtering. *Signal Processing Magazine, IEEE*, 20(5) :19–38, 2003. (Cité en page 70.)

[DM10] Adrien Delaye and Harold Mouchère. Vers une approche générique pour la reconnaissance de formes manuscrites structurées : application aux équations mathématiques et aux caractères chinois. In *Actes du Colloque International Francophone sur l'Ecrit et le Document (CIFED 2010)*, pages 297–312, 2010. (Cité en page 12.)

[Duf10] Richard Dufour. *Transcription automatique de la parole spontanée*. PhD thesis, Université du Maine, 2010. (Cité en page 47.)

[EB07] Cameron Elliott and Jeff A Bilmes. Computer based mathematics using continuous speech recognition. *Striking a C [h] ord : Vocal Interaction in Assistive Technologies, Games and More*, 2007. (Cité en page 51.)

[EDM+10] Yannick Esteve, Paul Deléglise, Sylvain Meignier, Simon Petitrenaud, Holger Schwenk, Loic Barrault, Fethi Bougares, Richard Dufour, Vincent Jousse, Antoine Laurent, et al. Some recent research work at lium based on the use of cmu sphinx. In *les actes de CMU SPUD Workshop, Dallas (Texas)*, 2010. (Cité en page 139.)

[EGS98] Y.A. Dimitriadis J.M. Cano Izquierdo J. J. Lopez Coronado B E. Gomez Sanchez, J.A. Gago Gonzalez. Experimental study of a novel neuro-fuzzy system for on-line handwritten unipen digit recognition. *Pattern Recognition Letters*, 19 :357– 364, 1998. (Cité en page 32.)

[Fao00] Nour-Eddin El Faouzi. Fusion de données pour l'estimation des temps de parcours via la théorie de l'évidence. *Recherche - Transports - Sécurité*, 68(0) :15 – 28, 2000. (Cité en page 57.)

[Fat98] Richard Fateman. How can we speak math. *Journal of Symbolic Computation*, 25(2), 1998. (Cité en pages 21, 44, 50 et 51.)

[Fat06] Richard J Fateman. Boxes, inkwells, speech and formulas draft. 2006. (Cité en page 50.)

[FDM98] Marcello Federico and Renato De Mori. Language modelling. *Spoken Dialogues with Computers*, pages 199–230, 1998. (Cité en page 140.)

[FS02] G.L. Foresti and L. Snidaro. A distributed sensor network for video surveillance of outdoor environments. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–525–I–528 vol.1, 2002. (Cité en page 67.)

[FTBM96] Richard J Fateman, Taku Tokuyasu, Benjamin P Berman, and Nicholas Mitchell. Optical character recognition and parsing of typeset mathematics. *Journal of Visual Communication and Image Representation*, 7(1) :2–15, 1996. (Cité en page 24.)

[Fur86] Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1) :52–59, 1986. (Cité en page 111.)

[FVGS09] Guihuan Feng, Christian Viard-Gaudin, and Zhengxing Sun. On-line hand-drawn electric circuit diagram recognition using 2d dynamic programming. *Pattern Recognition*, 42(12) :3215–3223, 2009. (Cité en page 19.)

[GAC+91] I. Guyon, P. Albrecht, Y. Le Cun, J. Denker, and W. Hubbard. Design of a neural network character recognizer for a touch terminal. *Pattern Recognition*, 24(2) :105 – 119, 1991. (Cité en page 33.)

[GB95] A. Grbavec and D. Blostein. Mathematics recognition using graph rewriting. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, volume 1 of *ICDAR '95*, pages 417–, Washington, DC, USA, 1995. IEEE Computer Society. (Cité en page 36.)

[GGY+11] Bharti W Gawali, Santosh Gaikwad, Pravin Yannawar, Suresh C Mehrotra, et al. Marathi isolated word recognition system using mfcc and dtw features. *ACEEE International Journal on Information Technology*, 1(1), 2011. (Cité en page 113.)

[GHLJ07] Yu-sheng Guo, Lei Huang, Chang-ping Liu, and Xin Jiang. An automatic mathematical expression understanding system. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 719–723. IEEE, 2007. (Cité en pages 38 et 39.)

[Gho12] Achraf Ghorbel. *Interprétation interactive de documents structurés : application à la rétroconversion de plans d'architecture manuscrits.* PhD thesis, INSA de Rennes, 2012. (Cité en page 19.)

[GJSF04] Cassandra Guy, Michael Jurka, Steven Stanek, and Richard Fateman. Math speak & write, a computer program to read and hear mathematical input. *University of California Berkeley*, 2004. (Cité en page 51.)

[GJT13] Sanjay Ghosh, Anirudha Joshi, and Sanjay Tripathi. Empirical evaluation of multimodal input interactions. In Sakae Yamamoto, editor, *Human Interface and the Management of Information. Information and Interaction Design*, volume 8016 of *Lecture Notes in Computer Science*, pages 37–47. Springer Berlin Heidelberg, 2013. (Cité en page 65.)

[GME11] Ben Gold, Nelson Morgan, and Dan Ellis. *Speech and audio signal processing : processing and perception of speech and music.* Wiley-Interscience, 2011. (Cité en pages 68 et 69.)

[GMS+03] M. Gandetto, L. Marchesooti, S. Sciutto, D. Negroni, and C.S. Regazzoni. From multi-sensor surveillance towards smart interactive spaces. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 1, pages I–641–4 vol.1, 2003. (Cité en page 72.)

[Gra06] Jean-François Grandin. Fusion de données – théorie et méthodes. *Techniques de l'ingénieur – Automatique avancée*, (ref. article : s7224), 2006. fre. (Cité en pages 57 et 60.)

[GSP+94] Isabelle Guyon, Lambert Schomaker, Réjean Plamondon, Mark Liberman, and Stan Janet. Unipen project of on-line data exchange and recognizer benchmarks. In *Pattern Recognition, 1994. Vol. 2-Conference B : Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 29–33. IEEE, 1994. (Cité en page 84.)

[Ham50] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2) :147–160, 1950. (Cité en page 40.)

[HCF+06] Jean-Paul Haton, Christophe Cerisara, Dominique Fohr, Yves Laprie, and Kamel Smaïli. *Reconnaissance automatique de la parole : Du signal à son interprétation.* Dunod, 2006. (Cité en pages 44, 46 et 111.)

[HCV04] Sandrine Henry, Estelle Campione, and Jean Véronis. Répétitions et pauses (silencieuses et remplies) en français spontané. *Actes des XXVème Journées d'Etude sur la Parole (JEP'2004)*, 2004. (Cité en page 109.)

[Her90] Hynek Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87 :1738, 1990. (Cité en page 47.)

[HHI07] Andreas Humm, Jean Hennebert, and Rolf Ingold. Modelling combined handwriting and speech modalities. In *Advances in Biometrics*, pages 1025–1034. Springer, 2007. (Cité en page 94.)

[HHP95] Jaekyu Ha, R.M. Haralick, and I.T. Phillips. Understanding mathematical expressions from document images. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 2, pages 956–959 vol.2, 1995. (Cité en page 29.)

[HNS04] Hartwig Holzapfel, Kai Nickel, and Rainer Stiefelhagen. Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3d pointing gestures. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 175–182. ACM, 2004. (Cité en page 65.)

[Hul96] Jesse F. Hull. Recognition of mathematics using a tow-dimensional trainable context-free grammar. Master's thesis, Massachuestts Institute Of Technology, 1996. (Cité en page 39.)

[IR10] Md Rabiul Islam and Fayzur Rahman. Hybrid feature and decision fusion based audio-visual speaker identification in challenging environment. *International Journal of Computer Applications*, 9(5) :9–15, 2010. (Cité en page 72.)

[Jel76] F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4) :532–556, 1976. (Cité en pages 44 et 112.)

[JNR05] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12) :2270–2285, 2005. (Cité en page 153.)

[KÖ3] Thomas Käster. Combining speech and haptics for intuitive and efficient navigation through image databases. In *Proc. International Conference on Multimodal Interfaces*, pages 180–187, 2003. (Cité en page 66.)

[KÏ2] Christine Kühnel. Evaluating multimodal systems. In *Quantifying Quality Aspects of Multimodal Interactive Systems*, T-Labs Series in Telecommunication Services, pages 13–21. Springer Berlin Heidelberg, 2012. (Cité en page 66.)

[Kai05] Edward C Kaiser. Shacer : a speech and handwriting recognizer. In *Proceedings of the International Conference on Multimodal Interfaces (ICMI), New York, NY, USA*, 2005. (Cité en page 94.)

[Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, (82 (Series D)) :35–45, 1960. (Cité en page 69.)

[Kal05] Slava Kalyuga. Prior knowledge principle in multimedia learning. *The Cambridge handbook of multimedia learning*, pages 325–337, 2005. (Cité en page 94.)

[KASA08] Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh, and Mo Nours Arab. Human-computer interaction : Overview on state of the art. *International Journal on Smart Sensing and Intelligent Systems*, 1(1) :137–159, 2008. (Cité en page 24.)

[KFDY01] Chan Kam-Fai and Yeung Dit-Yan. Error detection, error correction and performance evaluation in on-line mathematical expression recognition. *Pattern Recognition*, 34 :1671–1684, 2001. (Cité en page 39.)

[KGOI06] Kazutaka Kurihara, Masataka Goto, Jun Ogata, and Takeo Igarashi. Speech pen : predictive handwriting based on ambient multimodal recognition. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 851–860. ACM, 2006. (Cité en page 65.)

[KHDM98] J. Kittler, M. Hatef, R. P W Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3) :226–239, 1998. (Cité en pages 62 et 119.)

[KKKR13] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion : A review of the state-of-the-art. *Information Fusion*, 14(1) :28 – 44, 2013. (Cité en pages 58, 60, 65 et 66.)

[Kol02] A. S. Kolokolov. Signal preprocessing for speech recognition. *Autom. Remote Control*, 63(3) :494–501, March 2002. (Cité en page 110.)

[Kor08] András Kornai. Speech and handwriting. *Mathematical Linguistics*, pages 219–246, 2008. (Cité en page 94.)

[KRLK09] Kang Kim, Taik Heon Rhee, Jae Seung Lee, and Jin Hyung Kim. Utilizing consistency context for handwritten mathematical expression recognition. In *Tenth International Conference on Document Analysis and Recognition (ICDAR)*, pages 1051–1056, 2009. (Cité en page 39.)

[KRLP99] Andreas Kosmala, Gerhard Rigoll, Stephane Lavirotte, and Loic Pottier. On-line handwritten formula recognition using hidden markov models and context dependent graph grammars. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, pages 107–110. IEEE, 1999. (Cité en page 36.)

[KW07] Birendra Keshari and Stephen M Watt. Hybrid mathematical symbol recognition using support vector machines. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 859–863. IEEE, 2007. (Cité en page 34.)

[KW08] Birendra Keshari and Stephen M Watt. Online mathematical symbol recognition using svms with features from functional approximation. In *Electronic Proc. Mathematical User-Interfaces Workshop*, 2008. (Cité en page 34.)

[KWL95] M Koschinski, H-J Winkler, and M Lang. Segmentation and recognition of symbols within handwritten mathematical expressions. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 4, pages 2439–2442. IEEE, 1995. (Cité en page 24.)

[Lav00] Stéphane Lavirotte. *Reconnaissance structurelle de formules mathématiques typographiées et manuscrites*. PhD thesis, Université de Nice Sophia-Antipolis, 2000. (Cité en page 12.)

[LBOM98] Yann LeCun, Leon Bottou, GenevieveB. Orr, and Klaus-Robert Müller. Efficient backprop. In GenevieveB. Orr and Klaus-Robert Müller,

editors, *Neural Networks : Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 9–50. Springer Berlin Heidelberg, 1998. (Cité en page 199.)

[LBR+04] James Llinas, Christopher Bowman, Galina Rogova, Alan Steinberg, Ed Waltz, and Frank White. Revisions the jdl data fusion model ii. Technical report, DTIC Document, 2004. (Cité en page 57.)

[LC87] Yann Le Cun. *Modèles connexionnistes de l'apprentissage*. PhD thesis, Université Pierre et Marie Curie, 1987. (Cité en page 199.)

[Leb91] Frank Lebourgeois. *Approche mixte pour la reconnaissance des documents imprimés*. PhD thesis, 1991. (Cité en page 32.)

[LeC] http ://fr.dreamstime.com/photos-libres-de-droits-cinq-graphismes-de-sens-contrôlés-par-le-cerveau-image15603058. (Cité en page 56.)

[Lev66] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8) :707–710, 1966. (Cité en page 52.)

[Liv] Livescribe. *Livescribe, Smartpen User Guide, version 2.5*. (Cité en page 82.)

[LLH08] James Llinas, Martin E Liggins, and David L Hall. *Handbook of Multisensor Data Fusion : Theory and Practice*. CRC Press, 2008. (Cité en page 57.)

[LM95] James A Landay and Brad A Myers. Interactive sketching for the early stages of user interface design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 43–50. ACM Press/Addison-Wesley Publishing Co., 1995. (Cité en page 30.)

[LMM+06] George Labahn, Scott MacLean, Marzouk Mirette, Ian Rutherford, David Tausky, Wolfram Decker, Mike Dewar, Erich Kaltofen, and Stephen Watt. Mathbrush : An experimental pen-based math system. *Challenges in Symbolic Computation Software*, (06271), 2006. (Cité en page 79.)

[LWL96] Stefan Lehmberg, H-J Winkler, and Manfred Lang. A soft-decision approach for symbol segmentation within handwritten mathematical expressions. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 6, pages 3434–3437. IEEE, 1996. (Cité en page 30.)

[LY96] Richard F Lyon and Larry S Yaeger. On-line hand-printing recognition with neural networks. In *Microelectronics for Neural Networks, 1996., Proceedings of Fifth International Conference on*, pages 201–212. IEEE, 1996. (Cité en page 33.)

[MAGB05] Sébastien Macé, Eric Anquetil, Elodie Garrivier, and Bruno Bossis. A penbased musical score editor. *In proceedings ICMC, Barcelona, Spain*, pages 415–418, 2005. (Cité en page 30.)

[Mar71] William A. Martin. Computer input/output of mathematical expressions. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, SYMSAC '71, pages 78–89, New York, NY, USA, 1971. ACM. (Cité en page 20.)

[MaR10] João Magalhães and Stefan Rüger. An information-theoretic framework for semantic-multimedia retrieval. *ACM Trans. Inf. Syst.*, 28(4) :19 :1–19 :32, November 2010. (Cité en page 69.)

[MBE10] Lindasalwa Muda, Mumtaj Begam, and I Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv :1003.4083*, 2010. (Cité en page 113.)

[MM05] JB Mena and JA Malpica. Color image segmentation based on three levels of texture statistical evaluation. *Applied mathematics and computation*, 161(1) :1–17, 2005. (Cité en page 68.)

[MMD+05] Iain McCowan, Darren Moore, John Dines, Daniel Gatica-Perez, Mike Flynn, Pierre Wellner, and Hervé Bourlard. On the use of information retrieval measures for speech recognition evaluation. *IDIAP (Institut Dalle Molle d'Intelligence Artificielle Perceptive), Martigny, Switzerland. IDIAP Research Report IDIAP-RR 04-73*, 2005. (Cité en page 52.)

[MMW04] Georg F Meyer, Jeffrey B Mulligan, and Sophie M Wuerger. Continuous audio–visual digit recognition using n-best decision fusion. *Information Fusion*, 5(2) :91–101, 2004. (Cité en page 71.)

[MO11] Richard Maclin and David W. Opitz. Popular ensemble methods : An empirical study. *CoRR*, abs/1106.0257, 2011. (Cité en page 198.)

[MP43] Warren Sturgis McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115 – 133, 1943. (Cité en pages 32, 56 et 197.)

[MS03] Stephen Montgomery-Smith. Natural math, 2003. (Cité en page 51.)

[MV98] Erik G Miller and Paul A Viola. Ambiguity and constraint in mathematical expression recognition. In *Proceedings of the national conference on artificial intelligence*, pages 784–791. John Wiley & Sons LTD, 1998. (Cité en pages 14, 36, 38 et 39.)

[MVGG+11] Harold Mouchère, Christian Viard-Gaudin, Utpal Garain, Dae Hwan Kim, and Jin Hyung Kim. CROHME2011 : Competition on Recognition of Online Handwritten Mathematical Expressions. In *Proceedings of the 11th International Conference on Document Analysis and Recognition*, pages –, Beijing, Chine, September 2011. (Cité en pages 38, 39 et 78.)

[MVGK+12] Harold Mouchère, Christian Viard-Gaudin, Dae Hwan Kim, Jin Hyung Kim, and Utpal Garain. ICFHR 2012 - Competition on Recognition of

On-line Mathematical Expressions (CROHME 2012). In *Proceedings of ICFHR 2012*, pages 1–6, Bari, Italie, 2012. (Cité en pages 38, 39 et 78.)

[MVGZ+13] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, Utpal Garain, Dae Hwan Kim, and Jin Hyung Kim. Icdar 2013 crohme : Third international competition on recognition of online handwritten mathematical expressions. In *Proceedings of the 12th International Conference on Document Analysis and Recognition*, 2013. (Cité en pages 38, 39, 40 et 78.)

[MWS+09] Florian Metze, Ina Wechsung, Stefan Schaffer, Julia Seebode, and Sebastian Möller. Reliable evaluation of multimodal dialogue systems. In JulieA. Jacko, editor, *Human-Computer Interaction. Novel Interaction Methods and Techniques*, volume 5611 of *Lecture Notes in Computer Science*, pages 75–83. Springer Berlin Heidelberg, 2009. (Cité en page 66.)

[Nim04] Vincent Nimier. Information evaluation : A formalisation of operational recommendations. In *International Conference on Information Fusion*, pages 1166–1171, 2004. (Cité en page 66.)

[NMS+00] C. Neti, B. Maison, A. Senior, G. Iyengar, P. Decuetos, S. Basu, and A. Verma. Joint processing of audio and visual information for multimedia indexing and human-computer interaction. In *International Conference RIAO*, 2000. (Cité en page 70.)

[OIT01] Masayuki Okamoto, Hiroki Imai, and Kazuhiko Takagi. Performance evaluation of a robust method for mathematical expression recognition. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 121–128. IEEE, 2001. (Cité en page 38.)

[OM92] Masayuki Okamoto and Akira Miyazawa. An experimental implementation of a document recognition system for papers containing mathematical expressions. In HenryS. Baird, Horst Bunke, and Kazuhiko Yamamoto, editors, *Structured Document Image Analysis*, pages 36–53. Springer Berlin Heidelberg, 1992. (Cité en page 29.)

[Ovi03] S. Oviatt. User-centered modeling and evaluation of multimodal interfaces. *Proceedings of the IEEE*, 91(9) :1457–1468, 2003. (Cité en page 66.)

[PA03] Beryl Plimmer and Mark Apperley. Software to sketch interface designs. In *Proceedings of the ninth international conference on human-computer interaction (INTERACT'03), Zürich, Switzerland*, pages 73–80, 2003. (Cité en page 30.)

[PCT04] I. Potamitis, Huimin Chen, and G. Tremoulis. Tracking of multiple moving speakers with multiple microphone arrays. *Speech and Audio*

Processing, IEEE Transactions on, 12(5) :520–529, 2004. (Cité en page 69.)

[Pfl04] Norbert Pfleger. Context based multimodal fusion. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 265–272, New York, NY, USA, 2004. ACM. (Cité en page 71.)

[Pfl05] Norbert Pfleger. FADE - An Integrated Approach to Multimodal Fusion and Discourse Processing. In *Dotoral Spotlight at ICMI 2005*, 2005. (Cité en page 71.)

[PH+03] Berthille Pallaud, Sandrine Henry, et al. Amorces de mots et répétitions : des hésitations plus que des erreurs en français parlé. *Le poids des mots*, pages 848–858, 2003. (Cité en page 109.)

[PJME10] S. Petitrenaud, V. Jousse, S. Meignier, and Y. Estève. Speaker identification using belief functions. In *Information Processing and Management of Uncertainty (IPMU'10)*, Dortmund (Germany), 28 june - 2 july 2010. (Cité en page 63.)

[PKPM06] Vassilis Pitsikalis, Athanassios Katsamanis, George Papandreou, and Petros Maragos. Adaptive multimodal fusion by uncertainty compensation. In *INTERSPEECH*, 2006. (Cité en page 68.)

[Poi05] Emilie Poisson. *Architecture et apprentissage d'un système hybride neuro-markovien pour la reconnaissance de l'écriture manuscrite enligne*. PhD thesis, Ecole polytechnique de l'Université de Nantes, 2005. (Cité en pages 33, 100, 101, 102 et 103.)

[QMS+11] Solen Quiniou, Harold Mouchère, Sebastián Peña Saldarriaga, Christian Viard-Gaudin, Emmanuel Morin, Simon Petitrenaud, and Sofiane Medjkoune. Hamex-a handwritten and audio dataset of mathematical expressions. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 452–456. IEEE, 2011. (Cité en page 79.)

[Rab89] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989. (Cité en pages 34, 48 et 113.)

[Ram89] SR Ramesh. A generalized character recognition algorithm : a graphical approach. *Pattern Recognition*, 22(4) :347–350, 1989. (Cité en page 32.)

[Ram98] TV Raman. *Audio system for technical readings*, volume 1410. Springer Verlag, 1998. (Cité en page 51.)

[Rav10] Kumar Ravinder. Comparison of hmm and dtw for isolated word recognition system of punjabi language. In Isabelle Bloch and Jr. Cesar, RobertoM., editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 6419 of *Lecture Notes in Computer Science*, pages 244–252. Springer Berlin Heidelberg, 2010. (Cité en page 113.)

[RCC10] Ana Rebelo, G Capela, and Jaime S Cardoso. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(1) :19–31, 2010. (Cité en page 19.)

[RF03] A.F.R. Rahman and M.C. Fairhurst. Multiple classifier decision combination strategies for character recognition : A review. *Document Analysis and Recognition*, 5(4) :166–194, 2003. (Cité en pages 58, 62 et 119.)

[RK09] Taik Heon Rhee and Jin Hyung Kim. Efficient search strategy in structural analysis for handwritten mathematical expression recognition. *Pattern Recognition*, 42(12) :3192 – 3201, 2009. (Cité en pages 19, 32 et 97.)

[Ros96] Fabrice Rossi. Second Differentials in Arbitrary Feed-Forward Neural Networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume I, pages 418–423, Washington (USA), 6 1996. IEEE. (Cité en page 199.)

[Rou94] C. Rouchouze. Fusion de données : exemples défense et axes de recherche. *Traitement du signal*, 11(6) :459 – 464, 1994. (Cité en page 57.)

[RP97] V. Radova and J. Psutka. An approach to speaker identification using multiple classifiers. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1135–1138 vol.2, 1997. (Cité en page 71.)

[RS75] Lawrence R Rabiner and Marvin R Sambur. An algorithm for determining the endpoints of isolated utterances. *The Bell System Technical Journal*, 54(2) :297–315, 1975. (Cité en page 110.)

[RSF+99] Fukuda Ryoji, I Sou, Tamari Fumikazu, Ming Xie, and Suzuki Masakazu. A technique of mathematical expression structure analysis for the handwriting input system. In *Fifth International Conference on Document Analysis and Recognition*, pages 131–134, 1999. (Cité en pages 36, 38 et 39.)

[RSRS00] Mosur Ravishankar, Rita Singh, Bhiksha Raj, and Richard M Stern. The 1999 cmu 10x real time broadcast news transcription system. In *Proc. DARPA Workshop on Automatic Transcription of Broadcast News*. Citeseer, 2000. (Cité en pages 49 et 139.)

[RSTS06] Yamamoto Ryo, Sako Shinji, Nishimoto Takuya, and Sagayama Shigeki. On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. In *tenth International Workshop on Frontiers in Handwriting Recognition*, pages 249–254, 2006. (Cité en pages 37, 38, 39 et 97.)

[RZD00] Xiangshi Ren, Gao Zhang, and Guozhong Dai. An experimental study of input modes for multimodal human-computer interaction. In *Ad-*

vances in Multimodal Interfaces, ICMI 2000, pages 49–56. Springer, 2000. (Cité en page 66.)

[SB97] Holger Schwenk and Yoshua Bengio. Adaboosting neural networks : Application to on-line character recognition. In *Artificial Neural Networks–ICANN'97*, pages 967–972. Springer, 1997. (Cité en page 198.)

[SBW99] Alan N Steinberg, Christopher L Bowman, and Franklin E White. Revisions to the jdl data fusion model. In Belur V. Dasarathy, editor, *AeroSense'99*, volume 3719, pages 430–441. International Society for Optics and Photonics, SPIE, 1999. (Cité en page 57.)

[SC71] Hiroaki Sakoe and Seibi Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of the Seventh International Congress on Acoustics*, volume 3, pages 65–69, 1971. (Cité en pages 113 et 203.)

[SGH95] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time-delay neural networks and hidden markov models. *Machine Vision and Applications*, 8(4) :215–223, 1995. (Cité en page 34.)

[Sha76] Glenn Shafer. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976. (Cité en page 63.)

[SK94] Philippe Smets and Robert Kennes. The transferable belief model. *Artificial Intelligence*, 66(2) :191 – 234, 1994. (Cité en pages 63 et 64.)

[SK97] Bong-Kee Sin and Jin H. Kim. Ligature modeling for online cursive script recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6) :623–633, 1997. (Cité en page 34.)

[SLS07] Yu Shi, HaiYang Li, and Frank K Soong. A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 854–858. IEEE, 2007. (Cité en page 38.)

[Sme93] Philippe Smets. Quantifying beliefs by belief functions : an axiomatic justification. In *IJCAI*, volume 93, pages 598–603, 1993. (Cité en page 63.)

[Sme98] Philippe Smets. Probability, possibility, belief : which and where. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, 1 :1–24, 1998. (Cité en pages 63 et 64.)

[SN10] Marcos Serrano and Laurence Nigay. A wizard of oz component-based approach for rapidly prototyping and testing input multimodal interfaces. *Journal on Multimodal User Interfaces, Springer Publ.*, 3(3) :215–225, 2010. (Cité en page 66.)

[SNA99] Steve Smithies, Kevin Novins, and James Arvo. A handwriting-based equation editor. In *Proceedings of the 1999 conference on Graphics interface '99*, pages 84–91, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. (Cité en pages 28 et 29.)

[SRM12] Bhupinder Singh, Vanita Rani, and Namisha Mahajan. Preprocessing in asr for computer machine interaction with humans : A review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(3), 2012. (Cité en page 110.)

[STA92] Stanag 2022 : Intelligence reports, north atlantic treaty organization (nato), 1992. (Cité en page 66.)

[SVNS06] Richa Singh, Mayank Vatsa, Afzel Noore, and Sanjay K Singh. Ds theory based fingerprint classifier fusion with update rule to minimize training time. *IEICE Electronics Express*, 3(20) :429–435, 2006. (Cité en page 71.)

[SW06] Elena Smirnova and Stephen M Watt. A pen-based mathematical environment mathink. *Ontario Research Centre for Computer Algebra (Orcca), Research Report*, pages 1–14, 2006. (Cité en page 32.)

[SWS05] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM, 2005. (Cité en page 58.)

[Tap04] Ernesto Tapia. *Understaning Mathematics : A system for the recognition of on-line handwritten mathematical expressions.* PhD thesis, Fachbereich Mathematik und Informatik der Freien Universit at Berlin, 2004. (Cité en pages 28 et 34.)

[TB11] P.J. Blignaut T.R. Beelders. *Speech Technologies*, 2011. (Cité en page 66.)

[TD38] François Thureau-Dangin. *Textes mathématiques babyloniens*, volume 1. Brill Archive, 1938. (Cité en page 11.)

[TJGD08] Sergey Tulyakov, Stefan Jaeger, Venu Govindaraju, and David Doermann. Review of classifier combination methods. In *Machine Learning in Document Analysis and Recognition*, pages 361–386. Springer, 2008. (Cité en page 58.)

[TM97] Juan Manuel Torres-Moreno. *Apprentissage et généralisation par des réseaux de neurones : étude des nouveaux algorithmes constructifs.* PhD thesis, Institut National Polytechnique de Grenoble, 1997. (Cité en page 198.)

[TMB09] Jean-Philippe Thiran, Ferran Marqus, and Herv Bourlard. *Multimodal Signal Processing : Theory and applications for human-computer interaction.* Academic Press, 2009. (Cité en page 58.)

[Tou90] David S. Touretzky, editor. *Advances in neural information processing systems 2*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. (Cité en page 102.)

[TR03] Ernesto Tapia and Raul Rojas. Recognition of on-line handwritten mathematical formulas in the e-chalk system. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR)*, pages 980–984, 2003. (Cité en page 28.)

[TR07] Ernesto Tapia and Raúl Rojas. A survey on recognition of on-line handwritten mathematical notation. *Freie Universitat Berlin, Institut fur Informatik, Germany*, 2007. (Cité en pages 24, 32 et 34.)

[TSMS01] Kenichi Toyozumi, Takahiro Suzuki, Kensaku Mori, and Yasuhito Suenaga. A system for real-time recognition of handwritten mathematical formulas. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 1059–1063. IEEE, 2001. (Cité en page 38.)

[UNS05] Seiichi Uchida, Akihiro Nomura, and Masakazu Suzuki. Quantitative analysis of mathematical documents. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(4) :211–218, 2005. (Cité en page 24.)

[Van03] Patrick Vannoorenberghe. Un état de l'art sur les fonctions de croyance appliquées au traitement de l'information. *Revue I3*, 3(2) :9–45, 2003. (Cité en pages 63 et 64.)

[VGBP01] J. Vermaak, M. Gangnet, A. Blake, and P. Pérez. Sequential monte carlo fusion of sound and vision for speaker tracking. In *The 8th IEEE International Conference on Computer Vision*, volume 1, pages 741–746, 2001. (Cité en page 70.)

[VHH08] Ba-Quy Vuong, Siu-Cheung Hui, and Yulan He. Progressive structural analysis for dynamic recognition of on-line handwritten mathematical expressions. *Pattern Recognition Letters*, 29(5) :647–655, April 2008. (Cité en page 28.)

[vL09] Joeri van Laere. Challenges for if performance evaluation in practice. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 866–873. IEEE, 2009. (Cité en page 65.)

[VT68] Harry L Van Trees. *Detection, estimation and modulation theory. 1, Detection, estimation and linear modulation theory*. John Wiley, 1968. (Cité en page 57.)

[WCB06] Shengli Wu, Fabio Crestani, and Yaxin Bi. Evaluating score normalization methods in data fusion. In *Information Retrieval Technology*, pages 642–648. Springer, 2006. (Cité en page 153.)

[WES+09] Ina Wechsung, Klaus-Peter Engelbrecht, Stefan Schaffer, Julia Seebode, Florian Metze, and Sebastian Möller. Usability evaluation of

multimodal interfaces : Is the whole the sum of its parts ? In J.A. Jacko, editor, *Human-Computer Interaction. Novel Interaction Methods and Techniques*, volume 5611 of *Lecture Notes in Computer Science*, pages 113–119. Springer, Heidelberg, 2009. (Cité en page 66.)

[Whi87] F. E. White. Data Fusion Lexicon, Joint Directors of Laboratories. Technical report, Naval Ocean Systems Center, 1987. (Cité en pages 57 et 58.)

[WHP+09] Angela Wigmore, Gordon Hunter, Eckhard Pflügel, James Denholm-Price, and Vincent Binelli. Using automatic speech recognition to dictate mathematical expressions : The development of the ¨talkmaths¨ application at kingston university. *Journal of Computers in Mathematics and Science Teaching*, 28(2) :177–189, 2009. (Cité en page 51.)

[WKYJ03] Jun Wang, Mohan S Kankanhalli, Weiqi Yan, and Ramesh Jain. Experiential sampling for video surveillance. In *First ACM SIGMM international workshop on Video surveillance*, IWVS '03, pages 77–86, New York, NY, USA, 2003. ACM. (Cité en page 67.)

[WLH00] Yao Wang, Zhu Liu, and Jin-Cheng Huang. Multimedia content analysis-using both audio and visual clues. *Signal Processing Magazine, IEEE*, 17(6) :12–36, 2000. (Cité en page 69.)

[WVDM96] Alex Waibel, Minh Tue Vo, Paul Duchnowski, and Stefan Manke. Multimodal interfaces. *Artificial Intelligence Review*, 10(3-4) :299–319, 1996. (Cité en page 94.)

[YJYJ11] Li Yang, Le Jing, Yang Yuxiang, and Wang Jian. Improvement algorithm of dtw on isolated-word recognition. In *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, volume 3, pages 319–322, 2011. (Cité en page 113.)

[You06] S. Young. *The HTK Book*, université de cambridge edition, 2006. (Cité en page 51.)

[YP93] Liping Yang and Ramjee Prasad. Online recognition of handwritten characters using differential angles and structural descriptors. *Pattern Recognition Letters*, 14(12) :1019 – 1024, 1993. (Cité en page 32.)

[ZB11] R. Zanibbi and D. Blostein. Recognition and retrieval of mathematical expressions. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(4) :1–27, 2011. (Cité en page 24.)

[ZMVG13] Richard Zanibbi, Harold Mouchère, and Christian Viard-Gaudin. Evaluating structural pattern recognition for handwritten math via primitive label graphs. In *DRR*, 2013. (Cité en page 40.)

[ZPM+11] Richard Zanibbi, Amit Pillay, Harold Mouchere, Christian Viard-Gaudin, and Dorothea Blostein. Stroke-based performance metrics for handwritten mathematical expressions. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 334–338. IEEE, 2011. (Cité en page 40.)

[ZSY09] Yang Zhang, Guangshun Shi, and Jufeng Yang. Hmm-based online recognition of handwritten chemical symbols. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1255–1259. IEEE, 2009. (Cité en page 19.)

[ZYC06] Qiang Zhu, Mei-Chen Yeh, and Kwang-Ting Cheng. Multimodal fusion using learned text concepts for image categorization. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 211–220. ACM, 2006. (Cité en page 73.)

Appendices

Annexe A

Réseau de neurones

Description

Comme nous l'avons brièvement présenté dans la section 2.4, les *RNA* sont une organisation en couche de neurones formels (figure A.1). Ces derniers ont un pouvoir séparateur entre deux classes de symbole [MP43, DHS01, CM02, Bis06]. Ce pouvoir, dû à leur fonctionnement, vient du fait qu'un neurone formel se comporte telle une fonction effectuant la somme pondérée de toutes ses entrées, laquelle somme est passée à travers une transformation pour produire la sortie finale. Plus précisément, tel qu'illustré sur la figure A.1b, un neurone j de la couche $(n_c + 1)$ reçoit en entrée soit les sorties des neurones précédents, de la couche n_c (si c'est un neurone des couches cachées ou de la couche de sortie), soit les données d'entrées (notées x_i ici). Ces données sont pondérées par les poids $w_{n_c,i,j}$, dits *poids synaptiques*, puis sommées au sein du neurone j, pour former le *potentiel synaptique* σ_j. En plus des données qui arrivent au neurone, il existe un *biais* qui joue le rÃ´le d'un seuil, avec une activité fixée à 1, qui rajoute un poids supplémentaire au niveau de chaque neurone. Le potentiel synaptique σ_j est par la suite passé à travers la fonction $f(.)$ du neurone qui est sa fonction d'*activation* (ou de *transfert*) pour produire la sortie finale de ce neurone y_j. Cette sortie va être soit propagée dans les neurones de la couche suivante soit elle correspond à une sortie finale (classe) du réseau. Différentes fonctions d'activation sont définies, on cite parmi elles, la fonction *sigmoïdale*[1], la fonction de *Heaviside*[2] ou encore la fonction *softmax*[3].

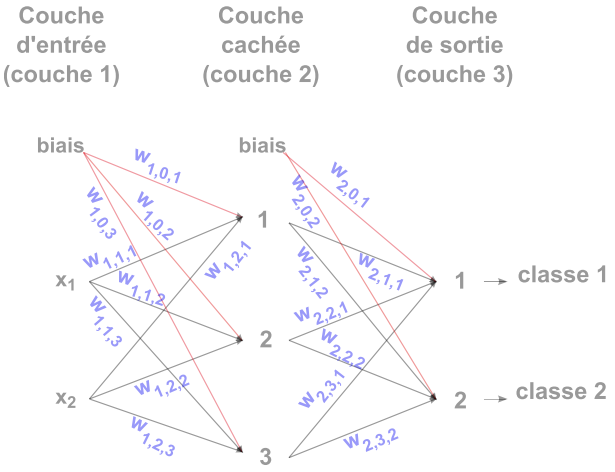
La mise en réseau des ces unités de base (neurones), tel que donnée en figure A.1a, permet de créer un classifieur capable de séparer plusieurs classes qui ne sont pas forcément linéairement séparables. Ce réseau va résumer la connaissance d'une base d'apprentissage en son sein, qu'il utilise par la suite pour accomplir sa tâche de reconnaissance.

Choix de la structure du réseau de neurones

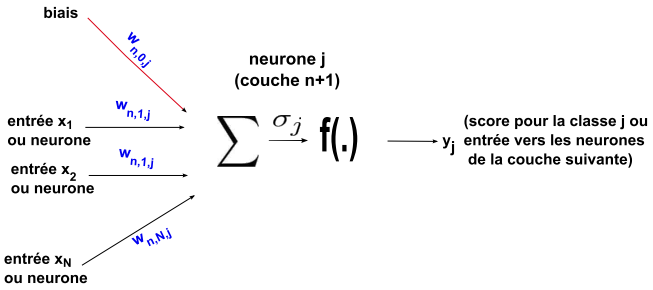
Il existe communément trois approche pour fixer de façon optimal la structure d'un réseau de neurones :

– La première est complètement empirique : plusieurs configurations différentes sont testées et la meilleure (celle se comportant au mieux face au problème

1. https ://en.wikipedia.org/wiki/Sigmoid_function
2. http ://en.wikipedia.org/wiki/Heaviside_step_function
3. https ://en.wikipedia.org/wiki/Softmax_activation_function



(a) Exemple de réseau de neurones simple : trois couches. La première est la couche d'entrée contenant deux neurones. La deuxième est une couche cachée contenant trois neurones et la dernière est la couche de sortie avec deux neurones



(b) Zoom sur le mode opératoire du neurone formel

FIGURE A.1 – Le neurone formel et le réseau de neurones

traité) est retenue, sachant qu'il n'existe pas de structure unique et optimale. Les mêmes performances peuvent être atteintes avec différentes topologies.

– Des approches, qui se veulent automatiques, utilisent des algorithmes de croissances, ou à contrario de dégénérescence, du réseau. Dans le premier cas, on part d'une structure minimale qui croît en fonction des performances. Dans le second cas, l'inverse est réalisé, c-à-d partir d'une structure complexe (surdimensionnée par rapport au problème) et la faire décroître [TM97].
– Une troisième variante, qui fait le lien entre les deux approches précédentes, existe. Il s'agit du méta-apprentissage, où c'est un autre réseau (le méta-réseau) qui est appris à faire une tâche de validation d'un autre réseau qui va s'occuper d'une tâche spécifique. En effet, le méta-réseau se charge de prendre en entrée la configuration du réseau que nous souhaitons configurer et en sortie, redonne les scores associés sans recourir au test de chacun des réseaux. C'est ces scores qui permettent de savoir quels réseaux peuvent éventuellement être optimaux et qui vont être testés [SB97, MO11].

Apprentissage du réseau de neurone

En effet, à un instant donné, si l'état du classifieur est donné par la matrice des poids courants W et si pour un vecteur de caractéristiques X_k d'un exemple k de la base d'apprentissage, le classifieur fournit en sortie la solution $Y(X_k, W)$, tout en sachant que la sortie désirée est $Y_d(X_k)$, alors faire l'apprentissage du système revient à traiter un problème d'optimisation par rapport aux poids W, soluble par une méthode de gradient. En effet, le but est de minimiser l'erreur globale $E(W)$ sur l'ensemble des exemples de la base d'apprentissage. Donc, cela revient à cumuler les erreurs sur chacun des exemples (comparer $Y(X_k, W)$ à $Y_d(X_k)$) de la base (voir équation A.1).

$$\begin{cases} E_k(W) = distance(Y(X_k, W), Y_d(X_k)), \ (erreur\ sur\ un\ exemple) \\ \\ E(W) = \sum_k E_k(W), \ (erreur\ totale\ sur\ toute\ la\ base) \end{cases} \tag{A.1}$$

L'optimisation citée précédemment, se base majoritairement sur des algorithmes de gradient (local ou global) qui s'avèrent être très performants [LC87, DHS01, CM02, Bis06]. La mise à jour des poids quant à elle se fait de façon itérative et, bien souvent, elle est basée sur le principe de rétropropagation de l'erreur [Ros96, LBOM98], méthode que nous avons également adoptée dans nos travaux. Dans ce type de configuration, une première propagation présente au classifieur l'exemple à classer. Une première estimation en résulte et permet d'opérer dans le sens inverse, en propageant, cette fois-ci, l'erreur au niveau local (de chaque neurone). Une dernière passe, sert à utiliser l'information extraite précédemment (les erreurs locales) pour mettre à jour les poids du réseau.

De façon formelle, si au début (réseau non entrainé) les poids initiaux sont $W^{(0)}$, et si les configurations des poids aux instants n et $n+1$, au niveau de la couche l du réseau et liant les neurones i et j, sont respectivement $W^{(n)}_{l,i,j}$ et $W^{(n+1)}_{l,i,j}$, alors l'état $n+1$ est déduit grâce au gradient global $\nabla E(W^{(n)}_{l,i,j})$ et est donné par l'équation A.2 :

$$W^{(n+1)}_{l,i,j} = W^{(n)}_{l,i,j} - \mu \times \nabla E(W^{n}_{l,i,j}) \tag{A.2}$$

μ étant le *pas d'apprentissage* ou le *pas de la descente*. Ce paramètre est très déterminant dans l'issue que peut prendre l'apprentissage. En effet, selon que sa valeur soit forte ou faible, la progression dans la courbe du gradient est différente : une valeur trop forte peu mener le système à osciller autour de la solution optimale sans jamais l'atteindre ; une valeur trop faible au contraire peut conduire à des temps d'apprentissage exorbitant mais surtout faire converger le système à un optimum local sans jamais pouvoir en sortir. C'est pour cela qu'un des choix judicieux est de prendre au début une valeur μ assez élevée pour accélérer la convergence, puis choisir une valeur faible qui va permettre l'affinement du modèle autour de l'optimum. En plus, afin d'éviter de tomber dans un optimum local, il convient de répéter ce schéma, même si en faisant cela, on risque de perdre temporairement en précision.

Le gradient global $\nabla E(W_{l,i,j}^{(n)})$ de l'équation A.2, quant à lui, est la somme des gradients locaux déduits des erreurs locales défini dans l'équation A.1. Il est donné par la formule A.3.

$$\begin{cases} \nabla E(W_{l,i,j}^{n}) = \sum_k \nabla E_k(W_{l,i,j}^{n}), avec\ : \\ \nabla E_k(W_{l,i,j}^{n}) = \dfrac{\partial distance(Y(X_k, W_{l,i,j}^{n}), Y_d(X_k))}{\partial W_{l,i,j}^{n}}. \end{cases} \tag{A.3}$$

Annexe B

Les systèmes à vastes marges (SVM)

Description

Pour faire suite à la courte description faite dans la section 2.4, le *SVM*, pour résumer la connaissance contenue dans une base de données concernant la distinction des $Nb_{Classes}$ classes du problème, va apprendre à délimiter les zones de chacune de ces classes (soit en les mettant les unes en vis-à-vis des autres : agir par couple de classe, ou en mettant systématiquement chaque classe contre toutes les autres, et cela, selon le méthode d'apprentissage adoptée). Ce faisant, deux contraintes pilotent cette démarche de recherche de meilleures frontières séparatrices. Il s'agit de maximiser la marge et de fixer la complexité des frontières. La marge concerne la distance des plus proches échantillons des classes à séparer (appelé **vecteurs de support**) à la frontière. La complexité, quant à elle, renseigne sur le nombre d'échantillons mal classés autorisés étant donné les frontières.

Dans le cas simple de séparation de deux classes, un hyperplan est défini par sont équation dans un espace de même dimension que celle des données à classer. L'équation B.1, donne l'exemple de l'hyperplan (H) séparant les classes C_1 et C_2 appliqué à une donnée (vecteur de caractéristiques) $\mathbf{x}$ à classer :

$$H(x) = \mathbf{w}.\mathbf{x} + \mathbf{b} \begin{cases} \geq 0 \ si \ C_1(y = 1). \\ \leq 0 \ si \ C_2(y = -1), \end{cases} \tag{B.1}$$

$\mathbf{w}$ (dans la même dimension que celle de $\mathbf{x}$) et b étant les coefficients définissant le plan H. y permet de décider de la classe résultante.

Trouver la marge maximale, revient au final à résoudre, l'équation sous contrainte B.2.

$$\begin{cases} minimiser \frac{1}{2}||\mathbf{w}||^2, \\ tel \ que \ : y \times H(x) = y \times (\mathbf{w}.\mathbf{x} + \mathbf{b}) \geq 1 \end{cases} \tag{B.2}$$

Ce problème est identifié comme étant un problème convexe sous contraintes linéaires de forme primale, dont il est question de minimiser la fonction objective [DHS01, CM02]. Sa résolution se passe après sa reformulation dans l'espace dual en appliquant le Lagrangien. Il s'agit, en fait, d'intégrer les contraintes au sein

de la fonction objective. L'équation B.3 donne cette formulation.

$$L(\mathbf{x}, \mathbf{b}, \alpha) = \frac{1}{2} - ||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i (y \times (\mathbf{w}.\mathbf{x} + b)) - 1, \tag{B.3}$$

avec les variables α_i qui sont les multiplicateurs de Lagrange. L doit être minimisé par rapport aux variables primales, les variables w et b, et maximisé relativement aux variables duales α_i.

Pour prendre en compte la non linéarité des frontières et/ou les exemples aberrants (*outlayers*), deux solutions sont apportées. La première concerne le relâchement des contraintes de l'équation B.2, en autorisant des erreurs de classification. Ceci rajoute une seconde contrainte sur la distance des ces exemples mal classés par rapport au plan optimal. En effet, le plan optimal va non seulement être celui qui va présenter une vaste marge possible, mais aussi celui qui minimise la distance des exemples mal classés au plan. Ceci introduit une variable supplémentaire $C > 0$, appelé paramètre de pénalisation de relâchement. Plus C est élevé, moins d'erreurs sont autorisées. Cette nouvelle formulation est donnée par l'équation B.4.

$$\begin{cases} minimiser \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i. \\ tel\ que\ : y_i \times (\mathbf{w}.\mathbf{x} + \mathbf{b}) \geq 1 - \xi_i, \end{cases} \tag{B.4}$$

avec $i = 1 \ldots n$, n étant le nombre d'exemples ; et $\xi_i \geq 0, \forall i$.

La seconde solution, concerne l'augmentation de la dimension de représentation. En effet, en faisant cela, des problèmes de classes non-linéairement séparables, le deviennent par ce passage à l'espace de représentation de dimension supérieure. Ce passage est généralement fait grâce à des fonctions non linéaires dites *fonctions noyaux*.

Annexe C

L'algorithme DTW

Description

L'algorithme DTW est basé sur de la programmation dynamique [SC71]. Il permet de réaliser un alignement entre deux signaux temporels, qui peuvent être de tailles différentes ou non. Le problème traité peut être vu comme la recherche du chemin le plus court dans le graphe qui représente les différentes associations possibles des échantillons des deux signaux à comparer. Dans le cas de la parole, c'est les vecteurs acoustiques ($MFCC$) de chacun des deux signaux qui vont être mis en correspondance. Pour décrire brièvement cet algorithme, supposons que nous disposons de deux séquences temporelles (les coefficients $MFCC$)) P et Q de tailles (nombre de fenêtres) m, n respectivement, données par l'équation C.1. Chaque point p_i (q_i) de P (Q) est représenté en dimension 39 (nombre de caractéristiques par point).

$$\begin{cases} P(m) = \{p_i \in \mathbb{R}^{39}; i \in \mathbb{N} | 1 \leq i \leq m\} \\ Q(n) = \{q_j \in \mathbb{R}^{39}; j \in \mathbb{N} | 1 \leq j \leq n\} \end{cases} \tag{C.1}$$

Trouver le chemin de déformation le plus court entre ces deux séquences, revient à trouver les associations de leurs points qui sont les moins couteuses en terme d'une certaine distance. En clair, si ce chemin est défini comme l'ensemble des couples de points mis en correspondance et est donné par $C(h) = \{\exists p_i \in P(m); \exists q_j \in Q(n) | (p_i(h), q_j(h)); i < h < H\}$, où $C(h)$ est le $h^{ième}$ couple impliquant les points $p_i(h)$ et $q_j(h)$. $C(1)$ et $C(H)$ sont respectivement les points de départ et d'arrivée du chemin. On donne sur la figure C.1, l'alignement d'un exemple de signal à classer comparé à deux exemples de la base de référence (lui même et un autre exemple).

D'après la figure C.1, quelque soit la forme du chemin C, ses points de départ ($C(1)$) et d'arrivée ($C(H)$) sont les mêmes. Cela est dû au fait que lors de la construction dudit chemin des conditions aux limites (ce qui permet d'avoir à chaque fois les mêmes points de départ et d'arrivée) et de continuité sont imposées. La condition aux limites assurent le fait d'utiliser la totalité des deux signaux à aligner et non seulement un segment (commencer des premiers points de chaque signal et finir par les points finaux des deux signaux). Cette condition est donnée par les formules de l'équation C.2.

$$\begin{cases} C(1) = (p_1, q_1) \\ C(H) = (p_m, q_n) \end{cases} \tag{C.2}$$

La condition de continuité, quant à elle, permet d'analyser l'ensemble des points des deux signaux. De ce fait, chaque point est au minimum utilisé une fois. Cela veut dire que lors de la construction du chemin, si on est à la position p_i, q_j, les prochains points à visiter sont forcément soit p_{i+1}, q_j, soit p_i, q_{j+1}, ou bien p_{i+1}, q_{j+1}. Donc, en opérant de la sorte, au sein des deux séquences, il n'y a pas de saut temporel lors du passage d'un point à un autre.

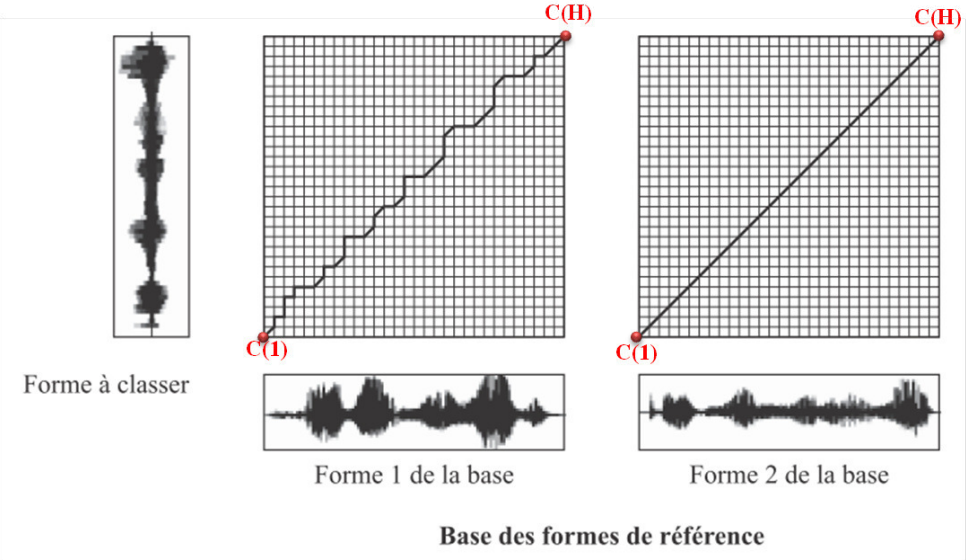


FIGURE C.1 – Exemple d'alignement d'un signal à classer avec deux des exemples de la base de référence grâce à la distance DTW [Bun97]

Le coût associé à un chemin est donné par la somme cumulée des distances entres les points des couples le constituant. Cette distance, notée $d(.,.)$ est dans notre cas une distance Euclidienne. Pour retrouver le chemin optimal et le coût qui lui est associé parmi tous les chemins possibles, de la programmation dynamique est utilisée en construisant une matrice de distances cumulées. Cette matrice implique tous les couples (p_i, q_j) et cela en choisissant à chaque fois la distance minimale parmi toutes les distances possibles (respecter la condition de continuité). La dernière valeur de cette matrice (dernière ligne, dernière colonne) représente le coût du chemin optimal, qu'on notera $D(P, Q)$. L'équation C.3 donne une présentation mathématique du propos rapporté dans ce paragraphe.

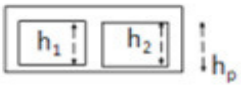
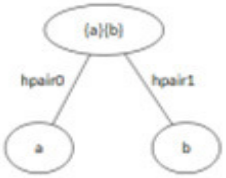
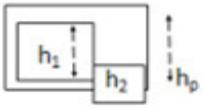
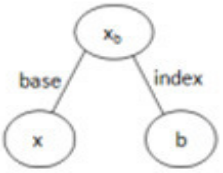
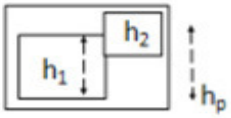
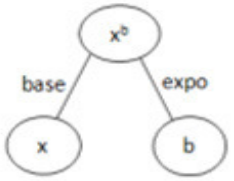
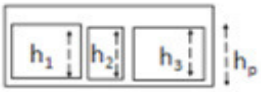
$$D(p_i, q_j) = d(p_i, q_j) + \min[D(p_{i-1}, q_j), D(p_i, q_{j-1}), D(p_{i-1}, q_{j-1})] \tag{C.3}$$

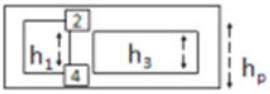
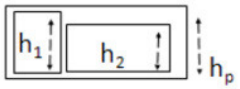
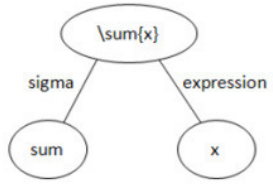
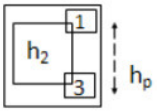
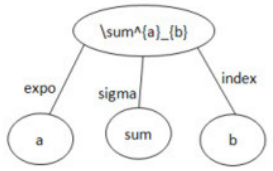
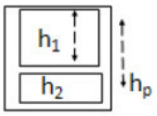
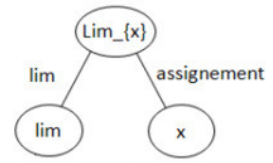
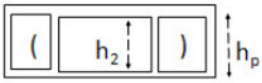
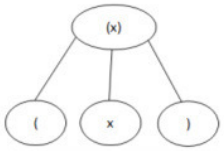
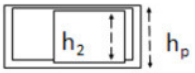
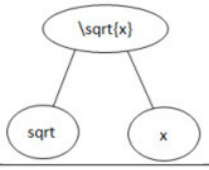
Le chemin optimal peut être retrouvé en faisant le chemin inverse ayant conduit au coût minimal (*backtracking*) et en sauvegardant les couples (p_i, q_j) impliquées à chaque fois.

Annexe D

Les relations spatiales considérées dans l'analyse structurelle

Dans cette annexe sont rapportées les descriptions précises des différentes relations spatiales utilisées dans notre système de reconnaissance des expressions mathématiques complètes telles qu'elles ont été définies dans [Awa10].

Relation	Sous-Relations	Représentation structurelle	Représentation en arbre
HPair	Hpair_left Hpair_right		
Subscript	Sub_base Sub_index		
Superscript	Sup_base Sup_expo		
Operator	Operator_ left Operator_ op Operator_ right		

Fraction	Frac_num Frac Frac_denom		
Integral	Int_intergral Int_expo Int_ind Int_ expression		
Sigma	Sigma_ sigma Sigma_ expression		
Sum	Sum_expo Sum_sigma Sum_index		
Lim	Lim_lim Lim_ assignement		
Parenthesis			
Sqrt			

l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Sofiane MEDJKOUNE

Stratégies de fusion pour des signaux écrits et sonores

Application à la reconnaissance d'expressions mathématiques

Résumé

L'être humain dans sa quête de mise en œuvre d'un dialogue le plus naturel possible avec sa machine s'inspire continuellement de la machine la plus perfectionnée connue à ce jour : l'être humain lui-même. Une caractéristique forte du dialogue entre humains est le recours à la multi-modalité. Le travail rapporté dans ce manuscrit porte sur l'étude, la conception et la validation d'un système de reconnaissance des expressions mathématiques, classe particulière de structures bidimensionnelles. Ce système est développé dans un cadre bimodal où l'on considère de façon complémentaire l'écriture manuscrite et la parole. La complémentarité qui existe entre ces deux modalités a été vérifiée et exploitée à profit dans notre système, d'abord dans un cadre simplifié qui est celui de la reconnaissance des symboles mathématiques isolés, puis dans un cadre plus général et plus réaliste, celui des expressions mathématiques complètes. La mise en place de ce système bimodal et sa validation requérant la disponibilité de données bimodales, nous avons collecté, complètement annoté et mis à disposition une base, nommée *HAMEX*, contenant 4350 expressions bimodales couvrant différents domaines. Nous montrons comment utiliser la transcription automatique de la dictée d'une expression pour guider le système de reconnaissance du signal manuscrit pour obtenir des résultats supérieurs au système monomodal. Les performances de ce système s'avèrent être meilleures que celles d'un système mono-modal basé uniquement sur le signal manuscrit seul.

Mots clés

Expression mathématique, reconnaissance de la parole, reconnaissance de l'écriture manuscrite, multi-modalité, fusion de données.

Abstract

Significant efforts are being done to make as natural as possible the way that human are interacting with their machines. Regarding this quest, a lot of research is being inspired by the most sophisticated machine ever known : human being and more precisely his use of the multi-modality aspect of the information to interact with his peers. The work reported here concerns the study, the conception and the validation of bidimensional structure recognition systems. The application considered here is the mathematical expression language which is one of the most interesting 2D languages. The system we proposed is original since it uses simultaneously two modalities to achieve its task. Indeed, both speech and handwriting streams are used by our system to perform the recognition in a bimodal fashion. This procedure allows dealing with the ambiguities arising when mono-modal processing is used. This system exploits the existing complementarity between the modalities in concern and exhibits an improvement of the performances with respect to the case of a mono-modal processing using only handwriting modality. To set-up, train and validate our system we built *HAMEX*, a bimodal database of mathematical expressions. This latter, is formed by 4350 mathematical expressions, each available in handwritten and audio forms and is fully annotated.

Key Words

Mathematical expression, speech recognition, handwriting recognition, multi-modality, data fusion.