

# AMI : Fondations Techniques

*Go Albert France. Centre R&D.*

10 février 2004

## Résumé

L'objectif de ce document est d'expliquer quelles sont les technologies utilisées dans le noyau des produits AMI et ce qui a procédé à leur choix. Il référence également d'autres technologies, utilisées notamment par des produits concurrents, et explique pourquoi elles ne sont pas, ou pas encore, utilisées dans AMI. Enfin, il doit permettre de fournir un vernis culturel permettant de répondre à des sollicitations extérieures sur ce que sont les fondations techniques des produits AMI et de les comparer aux solutions concurrentes.

Ce document n'aborde pas les solutions architecturales qui ont été mises en place pour les produits AMI. Elles sont plus du domaine du génie logiciel, même si ces solutions apportent également leur lot d'innovations, sur un plan purement informatique, et peuvent constituer de ce fait une plus-value commerciale.

.oOo.

## Table des matières

<b>1</b>	<b>Présentation</b>	<b>1</b>
1.1	Recherche de données et recherche d'informations . . . . .	1
1.1.1	Définitions . . . . .	1
1.1.2	Interprétation et pertinence . . . . .	2
1.2	Positionnement des produits AMI . . . . .	2
1.3	Composants d'un processus de recherche . . . . .	4
1.3.1	Le système documentaire . . . . .	4
1.3.2	L'index . . . . .	4
1.3.3	L'interface de recherche . . . . .	5
1.3.4	Fédération de plusieurs sources et fusion des résultats	5
1.3.5	Les outils de restitution des résultats . . . . .	7
1.4	Systèmes automatiques et systèmes manuels . . . . .	7
1.5	Traitement automatique de la langue naturelle . . . . .	8
1.5.1	Intérêt pour AMI . . . . .	8
1.5.2	Besoins et objectifs . . . . .	9
1.5.3	Dictionnaires et thésaurus . . . . .	10
1.6	Systèmes statiques et dynamiques . . . . .	10
<b>2</b>	<b>Taxonomie des modèles algorithmiques</b>	<b>12</b>
2.1	Le modèle booléen . . . . .	12
2.2	Le modèle vectoriel . . . . .	12
2.3	Le modèle probabiliste . . . . .	13
<b>3</b>	<b>Outils algorithmiques utilisés par AMI</b>	<b>14</b>
3.1	Modèle d'indexation . . . . .	14
3.2	Interprétation des requêtes . . . . .	15
3.3	Algorithmes d'identification du langage . . . . .	15
3.4	Algorithmes heuristiques . . . . .	16
3.5	Algorithmes linguistiques . . . . .	16
3.6	Agents coopératifs . . . . .	17
3.7	La technologie GMIL . . . . .	17
3.7.1	La forme de surface linguistique . . . . .	18

3.7.2	La sémantique distribuée . . . . .	19
3.7.3	Aspects innovants de GMIL . . . . .	19
3.8	Méthodes d'apprentissage . . . . .	19
3.8.1	Apprentissage par l'utilisation . . . . .	19
3.8.2	Apprentissage par l'environnement . . . . .	20
<b>A</b>	<b>Modèles vectoriels</b>	<b>21</b>
A.1	Modèle vectoriel classique . . . . .	21
A.2	Latent Semantic Indexing . . . . .	22
<b>B</b>	<b>Probabilités et réseaux bayésiens</b>	<b>23</b>
B.1	Origines . . . . .	23
B.2	Probabilité conditionnelle et formule de Bayes . . . . .	23
B.3	Réseaux bayésiens . . . . .	24
<b>C</b>	<b>Réseaux neuronaux artificiels</b>	<b>25</b>
C.1	Historique . . . . .	25
C.2	Modélisation . . . . .	25
C.3	Apprentissage et rétro-propagation . . . . .	26
<b>D</b>	<b>Logique floue</b>	<b>27</b>
D.1	Présentation . . . . .	27
D.1.1	Historique . . . . .	27
D.1.2	Imprécision et incertitude . . . . .	27
D.2	Définitions . . . . .	28
D.3	Caractéristiques d'un sous-ensemble flou . . . . .	28
D.4	Opérations sur les sous-ensembles flous . . . . .	29
D.5	$\alpha$ -coupes associées à un sous-ensemble flou . . . . .	31
D.6	Produit cartésien de sous-ensembles flous . . . . .	31
D.7	Relations et quantités floues . . . . .	31
<b>E</b>	<b>Théorie des possibilités</b>	<b>32</b>
E.1	Présentation . . . . .	32
E.2	Mesure de possibilité . . . . .	32
E.3	Mesure de nécessité . . . . .	33

## 1 Présentation

L'objet de ce document est une analyse technique des différentes technologies mises en œuvre dans le noyau AMI. Ce noyau est le cœur du savoir faire de tous les produits AMI. Ainsi, indépendamment des différentes utilisations d'AMI <sup>1</sup>, nous réduisons l'analyse, en guise d'illustration, au seul point de vue d'un système fonctionnel de recherche d'informations.

### 1.1 Recherche de données et recherche d'informations

#### 1.1.1 Définitions

Il existe une différence fondamentale entre la recherche de données (*Data Retrieval*) <sup>2</sup> et la recherche d'informations (*Information Retrieval*) [BYRN99].

La recherche de données se focalise sur l'extraction d'objets, au sens général, correspondant à des critères très précis (tels qu'une expression régulière ou une expression d'algèbre relationnelle) et pour laquelle une quelconque erreur représente un échec total. La recherche de données peut ne renvoyer aucune information, ce qui correspond à une information très précise indiquant qu'il n'existe rien relativement aux critères spécifiés. La recherche de données renvoie des résultats précis (l'ensemble des résultats renvoyés est homogène) qui peuvent d'ailleurs être quantifiables (décompte, cumul). La recherche de données est typiquement le domaine des systèmes de bases de données relationnelles pour lesquels l'information est fortement structurée et dont le modèle fournit la sémantique.

La recherche d'informations, quant à elle, s'établit sur des critères beaucoup plus souples qui incorporent notamment <sup>3</sup> :

- l'admission de critères vagues ou mal définis pour la recherche ;
- la tolérance d'erreurs dans les résultats renvoyés ;
- la possibilité de renvoyer des résultats dissemblables dans leur qualité et dans leur pertinence par rapport à la demande.

Cette recherche d'informations s'opère en règles générales sur des données qui sont structurées pour une compréhension humaine, à l'opposé des bases de données qui sont dédiées à des traitements mécaniques et difficilement accessibles aux êtres humains (pouvez-vous lire Oracle couramment?). Nous y trouvons donc évidemment le langage naturel, une sémantique ambiguë et une structure séquentielle difficilement appréhendable pour un système automatique. Concernant l'ambiguïté de la sémantique, le fossé est d'autant

---

<sup>1</sup>AMI propose plusieurs produits : *AMI Market Intelligence*, *AMI enterprise Discovery*, ... ([www.albert.com](http://www.albert.com))

<sup>2</sup>Nous fournirons la traduction en anglais de certains termes techniques afin d'aider la traduction de ce document en fournissant la correspondance exacte entre les termes. Les traductions doivent également permettre à un lecteur francophone un recoupement aisé avec la littérature anglo-saxonne.

<sup>3</sup>Nous pouvons déjà identifier ici les caractéristiques majeures d'un système flou, au sens de la logique floue, que nous aborderons plus en détail par la suite.

plus flagrant qu'un texte en langage naturel peut rester incompréhensible en dehors du contexte de son interprétation qui fait appel à la connaissance et à l'expérience du lecteur. Dans un système structuré, la formalisation du modèle implique généralement une clause de complétude qui désambiguise l'information et permet de s'assurer que l'information est exploitable.

### 1.1.2 Interprétation et pertinence

Le processus de recherche d'informations, qui cherche à satisfaire avant tout le besoin informatif de l'utilisateur humain, conduit à l'interprétation de la demande de l'utilisateur puisque cette demande est soit imprécise, soit ambiguë.

Cette interprétation de la demande n'est que la première étape du processus. Afin de pouvoir effectuer les recherches et de trouver des correspondances avec les documents recherchés, il est également nécessaire d'interpréter les documents, ce qui est souvent perçu comme l'extraction des informations syntaxiques et sémantiques des documents. La recherche de correspondance introduit alors naturellement la notion de mesure de pertinence (*relevancy*) afin de pouvoir quantifier la qualité des résultats en retour.

## 1.2 Positionnement des produits AMI

AMI se positionne résolument dans le domaine de la recherche d'informations. Il existe cependant une différence fondamentale entre AMI et les autres systèmes de recherche d'informations du marché : AMI se veut plus orienté vers les utilisateurs que vers le système documentaire.

La quasi-totalité des produits concurrents (à notre connaissance) se focalisent sur la représentation des documents et sur les techniques visant à améliorer cette représentation afin d'en extraire les éléments permettant une recherche pertinente. L'interface de recherche, très orientée *machine*<sup>4</sup>, est souvent le parent pauvre de l'application, son but n'étant que d'offrir l'accès à la circuiterie de l'outil de recherche.

L'approche dans AMI, à l'opposé, part du besoin de compréhension de la demande de l'utilisateur et cherche avant tout à apprendre à reconnaître ses besoins avant de procéder à la recherche documentaire.

Il s'en suit une division du système en plusieurs parties :

---

<sup>4</sup>C'est-à-dire que l'interface est destinée avant tout à satisfaire les besoins de l'application de recherche, ainsi qu'à l'exploitation de ses possibilités fonctionnelles, et se soucie fort peu de l'utilisateur. Cela nous donne par exemple les formulaires de *recherche étendue* des moteurs de recherche du Web qui imposent, en cas d'échec d'une recherche via l'interface simplifiée, l'utilisation d'un protocole complexe de spécifications de la requête : logique booléenne, définition de filtres et de critères d'exclusion, ajout ou retrait obligatoires de termes, etc.

- l’outil d’analyse de la requête et la reformulation de cette requête à destination des serveurs externes de recherche documentaires, via une ou plusieurs requêtes exprimées selon les protocoles et les capacités offerts par ces serveurs ;
- l’outil de référencement documentaire (l’index) propre à AMI qui permet de retrouver au mieux les informations recherchées.
- l’outil de fusion et de tri des résultats hétérogènes retournés par les divers serveurs (externes ou index AMI) sollicités, en vue de restituer à l’utilisateur une liste de résultats unifiée et informative.

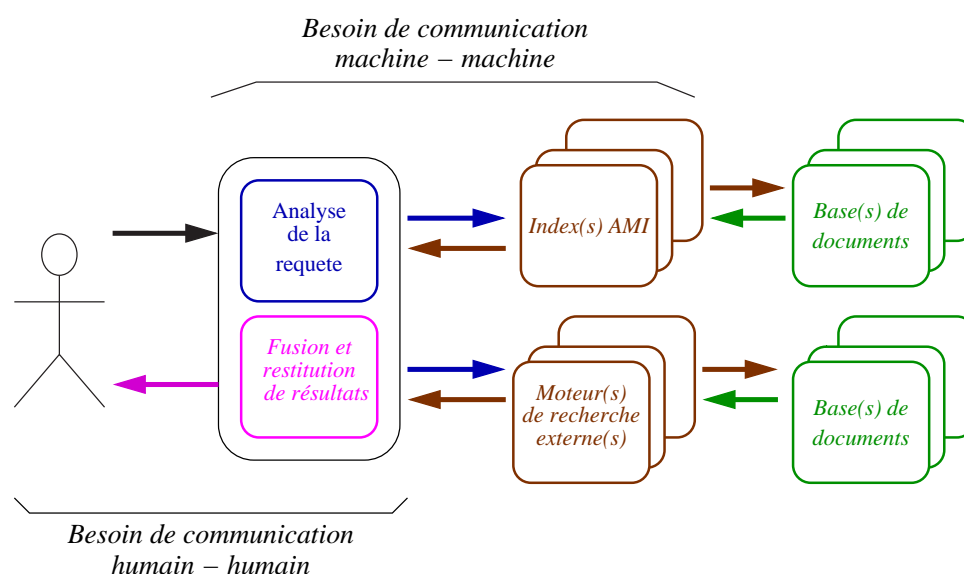


FIG. 1 – Différents composants d’un système de recherche AMI.

Cette modularité du système <sup>5</sup> permet en particulier de coupler le système d’analyse à un autre produit du marché, et ainsi de fédérer des recherches provenant de sources hétérogènes. C’est, aujourd’hui, une des caractéristiques fondamentales d’un système de recherche AMI qui le positionne dans le domaine des systèmes de recherche d’information distribuée<sup>6</sup>. Chacun de ces systèmes propose un choix plus ou moins important de sources à fédérer (des moteurs de recherche Web en général). Avec AMI, on dispose de sources spécifiques<sup>7</sup>, de sources type index AMI permettant un accès à une base documentaire intranet par exemple, et enfin on dispose d’un protocole de communication entre AMI et une source externe quelconque de données. Ce protocole permet aux intégrateurs d’AMI de mettre en œuvre très rapidement de

<sup>5</sup>À l’origine du produit, seule l’interface d’analyse de la requête était disponible. La nécessité de fournir un outil d’indexation est venue par la suite, lorsqu’il est apparu qu’une complémentarité analyse-indexation permettait de fournir des résultats encore meilleurs.

<sup>6</sup>Ce type de systèmes, communément appelés *MetaSearcher*, permettent via un seul point d’entrée de soumettre une requête utilisateur à plusieurs moteurs de recherche simultanément, et de fusionner les listes de résultats en retour. Des exemples de tels systèmes sont *Metacrawler* ([metacrawler.com](http://metacrawler.com)), *Copernic Agent* ([www.copernic.com](http://www.copernic.com)), ...

<sup>7</sup>Notamment un accès Web via le moteur de recherche *alltheweb*

nouvelles sources. Ainsi, AMI peut s'intégrer de manière non invasive dans un environnement applicatif des plus diversifiés. Cette dernière caractéristique confère à AMI un aspect innovant.

L'approche de recherche AMI intègre également une contrainte forte qui est celle d'éviter au mieux tout échec en retour, ce qui signifie que le système cherche toujours à répondre quelque chose, quel que soit la validité intrinsèque de ce retour, afin de ne pas placer l'utilisateur humain dans une situation d'échec et de rupture de communication. Cela autorise souvent une nouvelle formulation de la requête, ce qui constitue à la fois une nouvelle chance de trouver quelque chose de mieux et, dans le même temps, d'apprendre quelque chose de l'utilisateur à partir des associations linguistiques émises dans ses requêtes <sup>8</sup>.

### 1.3 Composants d'un processus de recherche

Il est important d'identifier les composants d'un processus de recherche afin de situer où sont utilisées les procédés algorithmiques d'AMI (ou ceux des produits concurrents).

#### 1.3.1 Le système documentaire

Le système documentaire (*Content Management System*) qui agit comme le gestionnaire du contenu. Cette partie est responsable de la réception, du stockage et de la restitution des documents. C'est également elle qui est en charge d'assurer les mécanismes de protection d'accès ou d'identification des utilisateurs, avec la définition possible de groupes. Un serveur Web (serveur HTTP en fait) est un exemple courant de gestionnaire de contenu. AMI ne fournit aucun gestionnaire de contenu et a pour vocation de pouvoir s'interfacer avec de tels systèmes.

#### 1.3.2 L'index

L'index est le mécanisme qui permet d'accéder rapidement aux documents. Par analogie totale avec les index des documents édités sur papier, l'index documentaire permet de retrouver rapidement les documents concernés sans avoir à parcourir séquentiellement et à analyser toute la masse documentaire. Techniquement, un index contient la liste inversée des références termes d'accès → documents. Cette partie est souvent dénommée *index inversé* (*inverted index*). L'index doit être bien sûr synchrone avec le gestionnaire de contenu. Si ce dernier est modifié dans le temps (ajout, retrait ou modification de documents), l'index doit être également mis à jour. Ce processus est appelé *politique d'indexation* (*crawling policy*).

<sup>8</sup>Il s'agit d'un des mécanismes mis en œuvre dans le service *AMI Query Expander* afin d'étendre une requête jugée trop pauvre et de ne l'étendre que par rapport au vocabulaire connu de l'utilisateur.

L'index est le composant qui résout le problème de la représentation du document pour sa recherche. Il s'agit du composant central sur lequel repose en général toutes les qualités d'un système de recherche d'informations. Les choix technologiques sur ce composant déterminent les performances intrinsèques du produit.

AMI fournit son propre produit d'indexation mais ne considère pas ce composant comme la partie centrale du système. Au contraire, il le décentre pour le mettre en face de la partie responsable de l'analyse de la requête (*interface de recherche*). Cette caractéristique unique permet d'ailleurs au système AMI de pouvoir coupler son moteur d'analyse à virtuellement n'importe quel autre système d'indexation, comme on peut le voir dans la représentation fournie en Figure 1.

### 1.3.3 L'interface de recherche

L'interface de recherche (*user interface*) est la partie en charge du dialogue avec l'utilisateur. Ces interfaces vont d'un formulaire très simple jusqu'à des procédés graphiques très perfectionnés qui ont pour principe de refléter la représentation documentaire [BYRN99, chap.10]. On peut distinguer dans cette interface deux parties complémentaires : a) une partie dialogue avec l'utilisateur qui autorise ce dernier à formuler sa demande, b) une partie dialogue avec le système d'indexation afin de récupérer les références documentaires (ou les documents eux-mêmes).

La complexité des systèmes concurrents à AMI est située dans la première partie, la seconde ne s'occupant que de la communication entre l'interface et l'index. AMI va à l'opposé en nécessitant une interface minimaliste (*l'interface zéro*) mais un processus de dialogue beaucoup plus complexe avec l'indexeur. L'objectif est de simplifier au maximum la complexité de communication entre l'utilisateur humain et la machine.

### 1.3.4 Fédération de plusieurs sources et fusion des résultats

Dans le cadre d'un processus de recherche d'information distribuée, la fédération de plusieurs sources de données hétérogènes nécessite le traitement d'une partie ou de la totalité des problèmes suivants[RHS03] :

- Sélection des sources de données appropriées à la demande de l'utilisateur, de manière à limiter la recherche à un ensemble réduit de sources et ainsi augmenter les performances (*server selection problem*)
- Reformulation de la demande de l'utilisateur dans le protocole approprié pour chaque source, puis, interprétation des résultats selon le format de retour propre à chacune.
- Fusion des résultats hétérogènes en une liste unifiée (*results merging problem*).

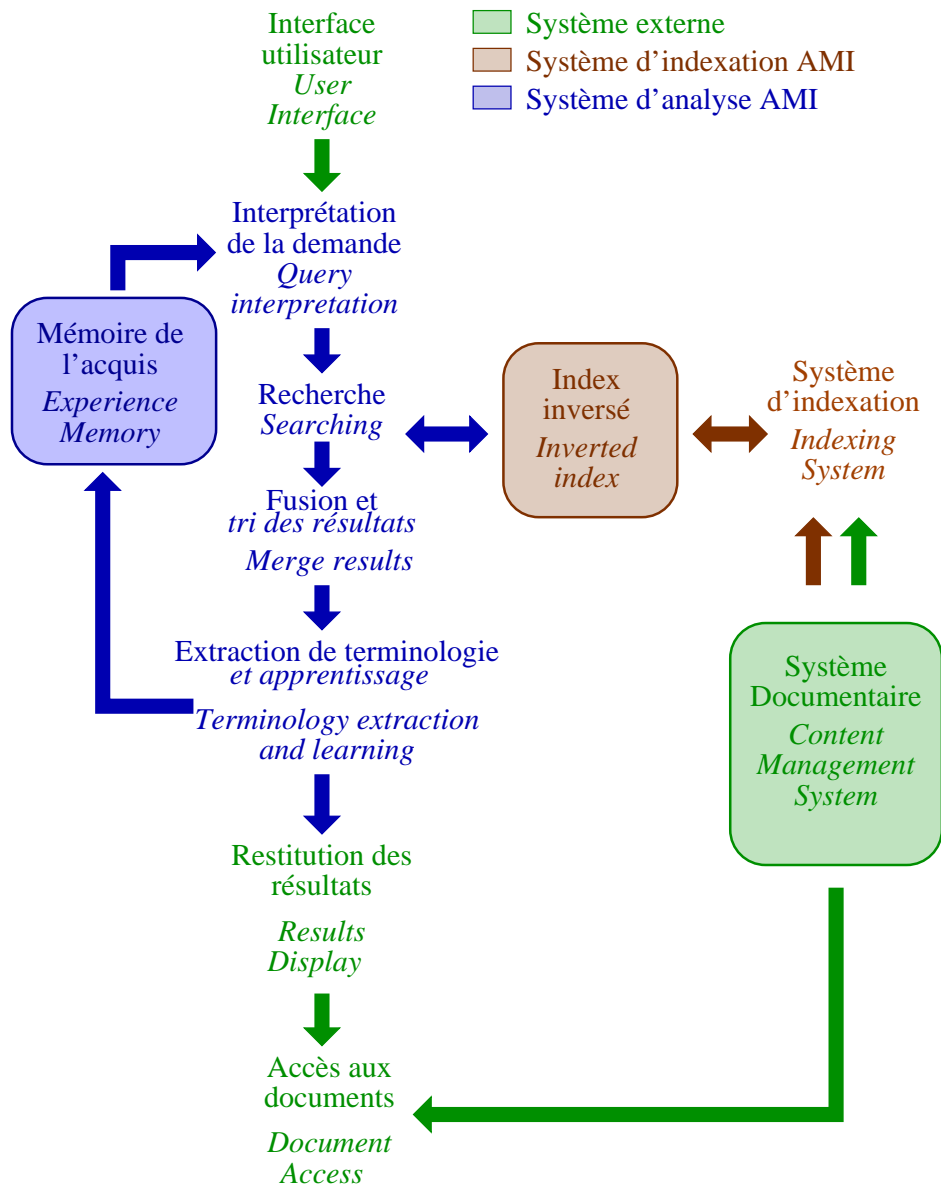


FIG. 2 – Processus de recherche d'informations.

La capacité du système de recherche AMI à pouvoir s'adresser simultanément à plusieurs index AMI ou index externes (moteurs de recherche) n'est pas reprise dans le schéma général représentant les différentes phases du processus de recherche (voir Figure 2) et ce, par souci de clarté.

AMI intègre des solutions pour les deux derniers problèmes.

La demande de l'utilisateur est enrichie après analyse. Cette forme enrichie est acheminée entièrement ou partiellement, vers les différentes sources, exploitant ainsi au mieux les capacités de leur langage de requête.

Concernant la fusion des résultats, plusieurs approches peuvent être utilisées [VGJL94][CHT99] :

- Imbrication des différentes listes (*round-robin strategy*) selon l'ordonnement fourni par les sources correspondantes.
- Trier les résultats selon les scores alloués par les sources (*raw score merging*)
- Recalculer les scores des résultats en fonction des différents éléments les décrivant et les trier en conséquence.

Les deux premières approches sont simplistes et ne peuvent donner de résultats satisfaisants sachant que chaque source possède son propre mode de calcul, qui est incomparable d'une source à l'autre. L'approche adoptée dans AMI est basée sur le calcul d'un score global pour chaque résultat en confrontant ces différents éléments descriptifs<sup>9</sup> à la requête enrichie<sup>10</sup>.

### 1.3.5 Les outils de restitution des résultats

Là encore, les méthodes de restitution des résultats varient grandement entre les systèmes [BYRN99, chap.10]. On y trouve trois grands groupes :

- le modèle *plat (flat)* qui restitue les références documentaires sous forme de liste ;
- le modèle guidé par la structure (*structure guided*) qui permet une visualisation (souvent graphique) fondée sur la structure de représentation du document ;
- le modèle hyper-texte (*hypertext*) qui fournit les références en utilisant les graphes générés par les liens hyper-textuels du schéma documentaire.

La position adoptée par AMI est de considérer cette partie comme un besoin externe, très dépendant de l'environnement requis par les utilisateurs finaux. La solution fournie est donc soit la restitution de résultats bruts (format XML ou autre), soit un habillage de présentation à partir de l'interprétation d'un modèle (*template*). Il s'agit en tout état de cause d'un modèle plat puisque le système se veut indépendant du système d'indexation (qui peut d'ailleurs être multiple, donc utiliser des structures internes très dissemblables).

## 1.4 Systèmes automatiques et systèmes manuels

L'une des caractéristiques fondamentales des systèmes de recherche d'informations est la méthode de constitution de l'index de recherche. Nous pouvons avoir :

---

<sup>9</sup>Les éléments descriptifs retournés par les sources sont habituellement un titre, un résumé et une liste de mots clés

<sup>10</sup>Les travaux décrits dans [RHS03] montrent qu'une stratégie basée sur un recalcul des scores de résultats à partir des éléments descriptifs, fournit des résultats équivalents en pertinence à un recalcul des scores à partir des textes entiers correspondant aux résultats

- Les systèmes *manuels* pour lesquels l'index est constitué manuellement dans son intégralité. Ces systèmes doivent bien sûr inclure des outils d'assistance à la constitution de l'index pour arriver à améliorer la productivité. Mais l'information contenue dans l'index est un résultat entièrement issu d'une analyse humaine des documents. Sa maintenance et sa mise-à-jour requièrent obligatoirement une intervention humaine.
- Les systèmes *mixtes* pour lesquels la majeure partie du travail de constitution de l'index est fait automatiquement par une machine, mais pour lequel il est nécessaire : soit de préparer en amont la masse documentaire, soit de calibrer en aval le comportement de l'application. Les deux tâches préparatoires ne sont pas mutuellement exclusives et elles requièrent l'intervention humaine. La maintenance et la mise à jour de tels systèmes nécessitent également une action humaine.
- Les systèmes *automatiques* pour lesquels tout est fait par la machine et dont l'intervention humaine est réduite initialement à son minimum. La maintenance et la mise-à-jour de l'index requièrent également une intervention humaine minimale.

Il est évident qu'aujourd'hui l'analyse d'un document, l'interprétation de son contenu et son rattachement à un système d'indexation sont des tâches infiniment mieux effectuées par un être humain que par une machine. Et ce quel que soit le niveau algorithmique mis en œuvre.

Par contre, la recherche d'une automatisation poussée dans le domaine de la recherche documentaire est motivée économiquement par le coût prohibitif que représente une "indexation humaine" <sup>11</sup>.

AMI se positionne définitivement comme un système automatique dans son fonctionnement nominal.

## 1.5 Traitement automatique de la langue naturelle

### 1.5.1 Intérêt pour AMI

Les domaines du TALN (Traitement Automatique de la Langue Naturelle) (NLP – *Natural Language Processing*) et de la recherche d'informations se côtoient depuis plus de 30 ans sans avoir lié de liens profonds sur le plan algorithmique. Ce malgré que la recherche d'informations puisse être considérée comme un sous-ensemble naturel du TALN [MS99, p.529] puisque l'interprétation des documents induit le traitement des langues <sup>12</sup>.

<sup>11</sup>Il existe un projet d'indexation du Web basé sur l'effort coopératif d'une communauté de responsables de domaines (*Open Directory* <http://dmoz.org/rdf.html>). L'idée est de fédérer le travail de fournis (gratuit !) d'une population importante et, par synergie, de fournir à tous un index de qualité. Pour séduisante qu'est l'idée, ce projet est cependant confronté à de nombreux problèmes, dont celui de la propriété intellectuelle des données accumulées.

<sup>12</sup>Avec l'extension des systèmes de recherche sur les données multimédia, cette inclusion devient caduque. Si elle demeure pour l'indexation de données vocales, elle ne l'est plus pour l'indexation de données purement sonores ou graphiques (images, vidéo). L'extension de la recherche d'information aux systèmes d'informations structurées éloigne également les deux

Récemment, K. Sparck Jones déclarait que les seuls domaines de la recherche d'informations pour lesquels le TALN pouvait avoir un impact sont ceux de *l'extraction d'information*, du *résumé automatique* et des *systèmes de question-réponse* [Ja00]. Il s'agit typiquement des trois domaines dans lesquels AMI se positionne.

### 1.5.2 Besoins et objectifs

Le texte reste le support privilégié des systèmes d'informations en dépit de l'utilisation intensive d'autre media. Cet aspect est renforcé par les bases de connaissances (littéraires, juridiques, commerciales) qui se développent sur l'Internet [CHL01].

Si l'analyse des textes est au cœur des préoccupations des deux domaines, les moyens d'y parvenir diffèrent par deux seuls critères : la rapidité et la qualité de cette analyse. Le TALN a tendance à privilégier la qualité de l'analyse au dépend de la rapidité. La recherche d'informations, dont le but est d'indexer de grands corpus de textes, privilégie plutôt la rapidité à une qualité linguistique.

L'approche qui a été adoptée dans AMI est de considérer que le système avait nécessairement besoin de se positionner sur les deux domaines, sans chercher à privilégier l'un par rapport à l'autre. Le besoin de prendre en compte le langage naturel est d'ailleurs un pré-requis aussi bien pour l'analyse de documents que pour permettre à l'utilisateur d'utiliser, dans sa communication avec la machine, l'outil qui lui est le plus familier, c'est-à-dire *sa* langue. Même si cela complique les besoins applicatifs, le langage naturel étant intrinsèquement humain et donc difficilement abordable pour la machine.

Cette approche hybride positionne AMI en retrait des outils purement linguistiques, dont il est d'ailleurs globalement inférieur en termes de performances linguistiques. Cette situation est motivée par deux objectifs :

- le besoin de rester efficace en termes de temps de traitement dans son analyse linguistique, pour l'indexation mais surtout pour l'analyse de la requête de l'utilisateur ;
- la tolérance à une construction linguistique des données qui peut, dans de nombreux cas usuels, ne pas respecter de manière absolue la syntaxe d'une langue et qui conduit souvent les outils dédiés du TALN à être en situation d'échec.

Ce dernier point nous amène souvent à dire que AMI *supporte* le langage naturel sans le requérir autrement qu'à des fins d'obtenir une information plus fine et plus riche. AMI n'est donc pas en situation de rejet si le texte ou la requête ne respecte pas rigoureusement (voire même pas du tout), la syntaxe du langage.

---

domaines, sauf pour l'interprétation de la requête de l'utilisateur pour lequel le langage naturel reste le meilleur véhicule d'expression.

AMI intègre des traitements linguistiques légers <sup>13</sup> pour l'analyse des requêtes. La réduction de la puissance des algorithmes est justifiée par le fait que les requêtes ne sont pas forcément exprimées en langage naturel et, d'autre part, que les requêtes sont presque toujours des textes très courts (voir §3.5 – p.16).

Dans la partie indexation, AMI intègre de l'algorithmique plus importante, les temps de traitement étant moins préoccupants que dans l'analyse des requêtes. La partie la plus innovante des algorithmes mis en place est la technologie GMIL (*Grammaire Minimaliste Indépendante du Langage*) (§3.7 – p.17)

### 1.5.3 Dictionnaires et thésaurus

La classification sémantique, comme tout traitement important sur le langage naturel, requiert la précision [Jon86, p.13]. C'est-à-dire la connaissance précise des éléments constitutifs du langage, sous la forme d'un dictionnaire ou d'un thésaurus.

AMI possède une autre approche sur ce point, en considérant notamment que la précision linguistique n'apporte plus beaucoup d'information au-delà d'un seuil de complexité de mise en œuvre des procédés requis pour atteindre cette précision.

Notamment, AMI se sert d'un lexique interne correspondant à sa mémoire, mais ne nécessite aucun dictionnaire extérieur, ni thésaurus.

## 1.6 Systèmes statiques et dynamiques

Un processus de recherche d'informations doit inclure au moins deux étapes qui sont :

- l'interprétation de la requête afin d'analyser au mieux ce que l'utilisateur recherche (*inférence* ou *déduction*) ;
- la mise en œuvre de l'analyse effectuée afin de procéder à la récupération des résultats (*application*).

On a alors une application statique au sens où, à système constant <sup>14</sup>, la même demande produira les mêmes résultats.

Cette staticité temporelle est classique pour les logiciels (c'est même un critère de stabilité d'une application) mais n'est absolument pas représentative de la pensée humaine pour laquelle la réception (consciente) d'information peut modifier l'état mémoriel (à court terme ou à long terme selon l'état émotionnel associé).

---

<sup>13</sup>Nous considérons la *légèreté* d'un algorithme comme une appréciation quantitative de la complexité algorithmique mise en œuvre et de sa rapidité à s'exécuter. Autrement dit, un traitement *léger* est un traitement simple et rapide.

<sup>14</sup>Un système constant est un système pour lequel l'environnement ne varie pas. Dans le cadre de la recherche d'informations, l'environnement est constitué de la base documentaire et de l'index.

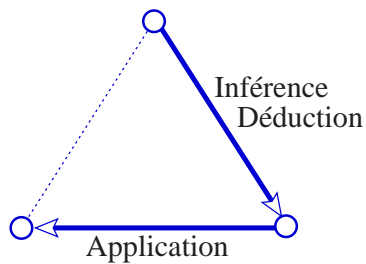


FIG. 3 – Cycle d'activité statique.

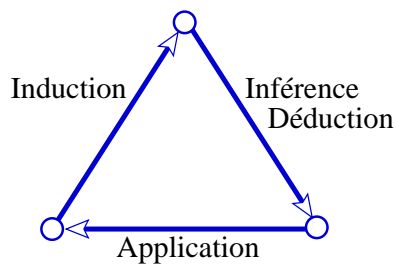


FIG. 4 – Cycle d'activité dynamique.

La transformation d'un système statique en un système dynamique est effectuée en ajoutant au cycle d'activité une étape supplémentaire (qui complète la boucle). Cette étape, l'*induction*, consiste à récupérer de l'information de l'action effectuée afin de transformer la mémoire du système et l'améliorer. On parle alors d'*expérience*.

Nous pouvons alors formaliser la notion d'*apprentissage* du système par : une application informatique est dite "capable d'apprendre" depuis une expérience  $E$  par rapport à un ensemble de tâches  $T$  et une mesure de performance  $P$  si ses performances à exécuter des tâches de  $T$ , mesurées par  $P$  sont améliorées par l'expérience  $E$  [Mit97, p.2].

En correspondance à cette définition, AMI peut être vu comme un système dynamique avec des capacités d'apprentissage. (voir §3.8 – p.19)

## 2 Taxonomie des modèles algorithmiques

Le marché des produits de recherche d'informations est peut être segmenté selon les fondements techniques qui sont sous-jacents à la représentation documentaire qui est implantée dans l'index.

Afin de pouvoir situer AMI dans ce contexte, ou tout au moins de s'y référer, nous expliquons brièvement les solutions qui correspondent à l'état de l'art en la matière.

### 2.1 Le modèle booléen

Dans le modèle booléen, qui a été l'un des tout premiers à voir le jour dans le domaine de la recherche d'informations, les requêtes et les documents sont vus comme des ensembles. Le processus de recherche s'effectue donc à partir d'une logique booléenne classique et sa formalisation bijective en matière de théorie ensembliste. Elle utilise les opérateurs d'intersection (ET), d'union (OU) et de complémentarité (négation).

Les limites d'un tel système, liées à sa simplicité, sont facilement visibles : la logique rigide d'appartenance ensembliste fait que la recherche d'un document utilisant les termes  $A$  et  $B$  doivent absolument contenir ces deux mots.

Ce modèle a engendré deux approches similaires :

- le modèle booléen étendu (*extended boolean*) qui est une méthode hybride s'approchant du modèle vectoriel puisqu'il considère les opérations booléennes en termes de distances algébriques [BYRN99, p.41] ;
- le modèle à base de sous-ensembles flous (*fuzzy sets*) qui assouplit le processus de recherche en prenant en compte la technique des sous-ensembles flous (voir §D – p.27) [BYRN99, p.34].

### 2.2 Le modèle vectoriel

L'origine des modèles vectoriels provient de l'insuffisance constatée de la logique binaire du modèle booléen. Ces modèles proposent alors de pouvoir avoir une correspondance partielle entre les termes au moyen d'une liste de poids non binaires : le *vecteur* descriptif d'un document ou d'une requête (voir A.1 – p.21).

L'application de l'algèbre vectorielle (puis matricielle) permet alors de modéliser la notion de similarité entre documents entre un document et une requête. Par extension, certains modèles (LSI par exemple) vont jusqu'à prendre en compte la décomposition matricielle en valeurs singulières et établir ainsi une identification calculatoire entre *valeur singulière statistique* et *concept* d'un document.

Les modèles complémentaires les plus courants sont :

- le modèle de vecteur généralisé (*generalized vector*) qui considère que les composantes du vecteur ne sont pas indépendantes<sup>15</sup>, donc que la base du système de coordonnées n'est pas orthogonale ;
- le modèle *Latent Semantic Index* (voir A.2 – p.22) ;
- le modèle à base de réseaux de neurones (voir C – p.25) dont on a cherché avant tout à utiliser les capacités en termes d'associations de motifs (*pattern matching*).

### 2.3 Le modèle probabiliste

Les modèles probabilistes tentent de résoudre les problèmes de recherche d'informations par une approche probabiliste. L'idée mise en œuvre est de considérer que les réponses aux questions sont des ensembles inconnus initialement mais dont on peut estimer la nature au moyen de pondérations probabilistes. Ensuite, grâce à l'utilisation du système et aux retours d'information de l'utilisateur final, le système ajuste ses pondérations afin de s'améliorer.

Les modèles probabilistes sont obligatoirement des systèmes dynamiques avec apprentissage.

Les modèles probabilistes courants sont :

- les réseaux bayésiens (*bayesian networks*) (voir B – p.23) ;
- les réseaux de croyance (*belief network*).

---

<sup>15</sup>Ce point est justifié par le fait que des termes différents dans un document ne sont pas indépendants, ce qui est une évidence sur le plan linguistique.

## 3 Outils algorithmiques utilisés par AMI

### 3.1 Modèle d'indexation

AMI utilise un schéma d'indexation de base fondé sur une variante du modèle vectoriel généraliste (voir §A.1 – p.21). Le choix a été effectué par un consensus entre :

- les besoins de simplicité des calculs d'indexation qui sont corrélativement liés à la notion d'efficacité en termes de temps d'indexation (i.e. plus de documents en moins de temps) ;
- la notion de *rendement* que l'on peut définir comme le besoin d'avoir une base algorithmique efficace mais qu'on ne cherche pas à développer au delà de ce que ses fondements théoriques autorisent.

Cette approche vise uniquement à choisir une modélisation de base et à considérer que la motivation initiale du modèle (notion de similarité) n'est qu'une représentation imparfaite de la réalité. En ce qui concerne le modèle vectoriel général, il est évident que la proximité des termes utilisés n'est ni une condition nécessaire, ni une condition suffisante pour que deux documents soient proches <sup>16</sup>.

Il est par contre possible d'améliorer ce modèle en créant un espace vectoriel "sémantique". Constitué de manière statistique (*Espace du mot de Schütze*) ou manuellement par la linguistique (comme le thésaurus utilisé par D. Yarowsky), la projection des termes sur cet espace permet d'en récupérer la sémantique et de calculer également une distance de proximité sémantique [Sch].

Il nous semble donc que ce modèle amélioré est une bonne approximation de la réalité d'autant que des masses documentaires regroupées dans une même base participent en règles générales à une uniformisation des procédés d'écriture :

- centres d'intérêt regroupés autour des mêmes thèmes ;
- nombre de rédacteurs limité, donc même référentiel culturel ;
- utilisation d'un vocabulaire dédié aux spécificités de la base documentaire.

De plus, il se trouve que ce modèle est toujours au moins aussi bon, voire meilleur, que les alternatives connues du modèle vectoriel [BYRN99, p.30]. C'est en particulier le cas pour la méthode LSI (voir §A.2 – p.22) dont la supériorité par rapport aux méthodes vectorielles classiques n'est pas prouvée [BYRN99, p.45] et dont le surcoût en matière de calculs n'est pas toujours justifié par les résultats obtenus [MS99, p.566].

<sup>16</sup>Deux exemples très courts. Les lexèmes "*association de malfaiteurs*" et "*bande de voyous*" sont proches sémantiquement mais n'ont aucun rapport au sens vectoriel, tout au moins sur la simple comparaison des termes entre deux (l'introduction d'une synonymie peut aider à améliorer les résultats dans ce cas. A contrario, les phrases "*le chien du fusil du chasseur assis*", "*le chasseur assis, son fusil et son chien*" et "*le chasseur assis en chien de fusil*" sont éloignées sémantiquement et pourtant très proches au sens vectoriel (là encore, l'introduction d'une reconnaissance des lexèmes complexes peut aider à la désambiguïsation).

L'indexation de base est complétée par la mise en œuvre de GMIL (voir §3.7 – p.17).

### 3.2 Interprétation des requêtes

L'interprétation des requêtes met en jeu trois grandes familles de procédés :

- les algorithmes d'identification du langage et de l'encodage utilisé qui permettent le traitement des différentes langues humaines ;
- les algorithmes heuristiques, indépendants de la langue, qui permettent une première approche de compréhension de la requête ;
- les algorithmes linguistiques qui appliquent des procédés plus fins, liés au langage, et qui permettent d'approfondir les analyses préliminaires.

L'indexation des documents fait appel également à l'ensemble des algorithmes afin de conserver une cohérence entre l'analyse de la requête et les méthodes d'accès à l'index inversé.

### 3.3 Algorithmes d'identification du langage

Une des ambitions d'AMI est d'être multi-lingue et universel. Les algorithmes d'identification du langage permettent alors de résoudre une double problématique :

- une reconnaissance de l'encodage utilisé dans le document <sup>17</sup> permettant ainsi de traiter virtuellement n'importe quelle langue ayant un support d'encodage informatique ;
- une reconnaissance de la langue utilisée au moyen d'algorithmes coopératifs (voir §3.6 – p.17).

La reconnaissance de l'encodage permet de transformer les textes des documents dans un encodage universel <sup>18</sup>, ce qui rend homogène l'indexation documentaire.

La reconnaissance de la langue est actuellement basée sur quatre algorithmes coopératifs qui sont :

- l'encodage utilisé pour la langue (certaines langues sont facilement identifiées par leur encodage, notamment les langues slaves, asiatiques ou sémitiques ; on pourrait néanmoins utiliser un encodage japonais pour écrire un texte entièrement en anglais ou en russe puisque cet encodage intègre les alphabets correspondants ; c'est néanmoins très rare dans la pratique) ;

---

<sup>17</sup>La reconnaissance de l'encodage d'une requête d'un utilisateur n'est pas réalisable de manière fiable. La raison essentielle est que le texte est très court, invalidant tout procédé statistique, et que sur quelques caractères les ambiguïtés entre certains encodages, notamment européens, sont énormes.

<sup>18</sup>Le système utilise en fait deux encodages universels. Le premier (UCS-2), à usage interne, est un codage fixe sur deux octets couvrant la quasi-totalité des besoins des langues modernes traitées par l'informatique (ne sont pas couverts : les hiéroglyphes, certains caractères asiatiques anciens, les écritures cunéiformes et quelques autres anciennetés). Le second (UTF-8) est un codage variable (de un à six octets), à usage externe, compatible avec les encodages US-ASCII et ISO-8859-1 (i.e. ISO-Latin 1).

- la reconnaissance de mots du langage ;
- l'utilisation de statistiques fondées sur les variations morphologiques des mots et sur des règles étymologiques ;
- la connaissance préalable de l'utilisateur et de ses habitudes.

Le dernier point est l'une des utilisations faites par le système du profil utilisateur.

Le système manipule une notion floue des langues permettant ainsi de conserver toute ambiguïté résiduelle en cas d'échec de toute ou partie de l'analyse.

### 3.4 Algorithmes heuristiques

Les algorithmes heuristiques sont les algorithmes, indépendants de la langue après découpage du document ou de la requête en termes linguistiques <sup>19</sup>, qui permettent une première approche d'analyse de la requête.

Ces algorithmes portent principalement sur :

- des procédés de recherche d'occurrences proches des unités linguistiques afin, soit de trouver de potentielles variations orthographiques, soit de trouver des termes substituables pour une interprétation ;
- des procédés d'associations de termes afin de détecter des lexèmes complexes ;
- l'exploitation de l'historique de l'utilisateur afin de pouvoir enrichir la recherche à partir d'un contexte identifié au préalable, ou prévoir la substitution de termes par une synonymie contextuelle.

Rentre également dans cette catégorie l'algorithme d'extension de la requête qui permet, dans le cas d'une formulation trop pauvre de la question <sup>20</sup>, d'enrichir la recherche par l'ajout de termes corrélés soit aux demandes précédentes de l'utilisateur, soit de termes issus d'autres utilisateurs partageant les mêmes centres d'intérêt.

Dans le cas d'une requête de recherche, les algorithmes heuristiques fournissent au système une "interprétation" de la question qui est constituée d'un graphe d'hypothèses.

### 3.5 Algorithmes linguistiques

Dans le cas où la langue est reconnue (c'est le cas de la quasi-totalité de textes longs ou des requêtes des utilisateurs formulées en langage naturel ; l'ambiguïté provient essentiellement des requêtes courtes sans structure syntaxique), le système met en œuvre des algorithmes linguistiques. Ces algo-

<sup>19</sup>La segmentation d'un texte en unités linguistiques est spécifique à la langue. Procédé très complexe pour les langues n'utilisant pas d'espace séparateur (comme les langues asiatiques), elle se simplifie pour les langues européennes mais conserve des règles qui sont propres à une langue.

<sup>20</sup>La pauvreté de la requête se mesure par la quantification de sa valeur sémantique. Cela inclut, entre autres, le nombre de mots significatifs de la requête.

rithmes appliquent des procédés plus fins, liés au langage, qui permettent d'approfondir les analyses.

Sont couverts par ces algorithmes : les problèmes de lemmatisation (*stemming*) ou racinisation, l'interprétation syntaxique ou sémantique légère ainsi que des règles spécifiques à chaque langue.

Dans le cas d'une requête de recherche, ces algorithmes produisent un enrichissement du graphe d'hypothèses.

### 3.6 Agents coopératifs

L'une des idées fondamentales mises en œuvre dans AMI est de considérer que l'interprétation humaine d'un contenu documentaire ne s'effectue pas au moyen d'un seul et unique algorithme, aussi complexe soit-il, mais via la coopération de mécanismes plus simples, plus spécifiques, mais aussi plus performants.

La synergie créée par cette coopération est simulée par l'activité d'agents logiciels spécialisés. Cette approche offre les avantages suivants :

- possibilité de gérer les ambiguïtés en confrontant les résultats antagonistes de procédés similaires ou corrélés ;
- possibilité de traiter les cas d'échec d'algorithmes en paliant leur insuccès par d'autres résultats ;
- amélioration des performances par la possibilité de mettre en jeu des composants plus rapides ;
- amélioration de la robustesse du système par la possibilité de mettre en jeu des composants plus simples.

Le système fait la synthèse des éléments d'information au moyen de la logique floue. L'une des règles adoptée est d'éviter de faire un choix (processus de *defuzzification*) tant qu'aucune contrainte ne l'y impose. On rejoint ainsi le comportement humain qui, par abstraction ou globalisation de l'information, arrive à obtenir un résultat sans avoir toute la précision des données nécessaires à l'action.

### 3.7 La technologie GMIL

Considérant le besoin de prendre en compte le langage naturel dans l'analyse des documents (voir §1.5 – p.8), AMI intègre une technologie spécifique et innovante appelée GMIL (*Grammaire Minimaliste Indépendante du Langage*). GMIL se positionne comme une solution hybride qui s'appuie fortement sur un modèle linguistique mais dont la légèreté des algorithmes TALN mis en œuvre l'autorise à :

- être rapide pour traiter de gros corpus de textes ;
- intégrer des algorithmes globalement indépendants de la langue du document.

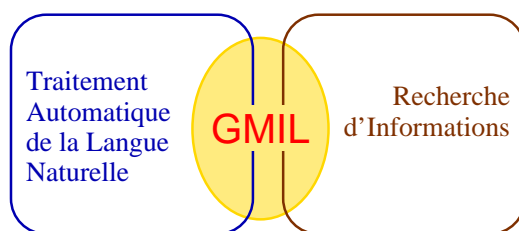


FIG. 5 – GMIL : solution hybride.

GMIL extrait d'un document (réduit à un texte structuré) une liste de termes ordonnés par un calcul statistique de fréquences. GMIL s'appuie sur une *forme de surface linguistique* du texte pour extraire des termes qu'il pondère de manière statistique, exprimant par cela la *sémantique distribuée* du document.

### 3.7.1 La forme de surface linguistique

La forme de surface linguistique du texte peut se définir communément comme une réunion de principes linguistiques, comme la forme (morphologie) des mots et leur place dans le texte (syntaxe).

On parle de forme de surface par opposition à la structure d'une analyse morpho-syntaxique profonde qui construit des arbres de dérivation syntaxique à partir d'une racine représentant le texte. Ces analyses morpho-syntaxiques sont trop longues en temps d'exécution, trop coûteuses en place mémoire et surtout trop sensibles à un non-respect de la syntaxe.

D'un autre côté, les extracteurs de surface des systèmes de recherche d'informations se contentent de prendre des mots ou des parties de mots (lemmatisation automatique, méthodes des  $n$ -grammes) afin de lancer des calculs statistiques sur cette masse de données. Ils ne tiennent pas compte, ou très peu, de modèles linguistiques.

GMIL tient compte d'un modèle linguistique dérivé d'une structure linguistique profonde en la restreignant à une forme de surface. Il recherche donc à définir des catégories syntaxiques les plus fines possibles – ce qui correspond à un traitement algorithmique profond – mais en adoptant une stratégie plus souple, donc plus rapide et plus tolérante. L'ensemble produit un résultat de qualité qui est proportionnel à la qualité syntaxique du texte analysé.

Au moyen de ce modèle de surface, GMIL extrait des *syntagmes nominaux* (SN) qui sont affinés en calculant des poids à partir des fréquences d'apparition des termes et des catégories syntaxiques. Ce calcul de fréquences d'apparition constitue la *sémantique distribuée*.

### 3.7.2 La sémantique distribuée

La *sémantique distribuée* (ou *distributionnelle*) est un terme issu de la recherche d'informations. Elle repose principalement sur une théorie de philosophie du langage, la *théorie de l'utilisation* (*use theory*) dont L. Wittgenstein en donne une définition [Wit68, §43] :

“Pour une grande classe de cas – mais pas tous – quand nous employons le mot *sens*, celui-ci peut être défini par : le sens d'un mot est celui de son utilisation dans le langage.”

Les méthodes de catégorisation automatique actuelles s'appuient sur cette base philosophique, notamment les modèles vectoriels avec les techniques *tf-idf* (§A.1 – p.21), le LSI (§A.2 – p.22) ou les réseaux de neurones (Espace du mot de Schütze) (§C – p.25). C'est aussi cette méthode qui est appliquée par GMIL à l'ensemble des syntagmes nominaux issus du texte.

### 3.7.3 Aspects innovants de GMIL

Les innovations de GMIL se situent sur deux plans <sup>21</sup> :

- l'implantation d'une méthode hybride entre le TALN et les méthodes de la sémantique distribuée, ce qui constitue une approche très peu commune aujourd'hui ;
- la mise en œuvre d'algorithmes linguistiques qui se fondent sur des prémisses linguistiques génériques aux langues (d'où le nom attribué à la technologie), les spécificités de chaque langue étant associées à ces prémisses par spécialisation.

## 3.8 Méthodes d'apprentissage

### 3.8.1 Apprentissage par l'utilisation

AMI met en œuvre, lors de son utilisation, des algorithmes d'apprentissage sur la base des requêtes reçues et des résultats récupérés à partir de ces requêtes.

La mémoire du système est une mémoire floue dans laquelle il ne faut pas voir un entrepôt de connaissances encyclopédiques mais une simple connaissance expérimentale de son environnement de travail. Autrement dit, le but fondamental de l'apprentissage dans AMI n'est pas d'apprendre et de déduire des vérités académiques mais de connaître les vérités relatives à son environnement et à ses utilisateurs.

L'apprentissage effectué à partir des résultats d'une recherche passe par la mise en œuvre d'un algorithme d'extraction de terminologie à partir de texte.

<sup>21</sup>La technologie GMIL fait actuellement l'objet d'une demande de brevet européen no. 02406526.9, auprès de ABREMA (Agence Brevets et Marques GANGUILLET & HUMPHREY)

Celui-ci est basé essentiellement sur des critères statistiques et linguistiques.

L'apprentissage qui en découle, concerne :

- le vocabulaire inconnu du système <sup>22</sup> ce qui constitue au fil du temps une spécialisation du vocabulaire liée à son utilisation ou à la découverte de néologismes ;
- le vocabulaire déjà connu mais enrichi, désambiguïsé ou, au contraire invalidé, ce qui permet au système de s'adapter aux variations sémantiques que les langues *vivantes* apportent à la signification des mots selon l'usage, les événements ou les modes ;
- la constitution de lexèmes complexes permettant une perception plus fine des associations de mots, noms composés, noms propres, etc.

L'apprentissage est aussi lié à la connaissance statistique de l'utilisateur (*user profiling*). Cette connaissance, parfaitement anonyme puisque basée sur des mécanismes d'identifiants numériques <sup>23</sup>, permet de mémoriser un historique des requêtes effectuées par l'utilisateur. Sur cette base, le système en déduit des données factuelles :

- langues utilisées ;
- vocabulaire utilisé et caractéristiques de ce vocabulaire (richesse, fautes potentielles ou courantes, etc.) ;
- associations entre les mots ou termes ;
- centres d'intérêts déduits du vocabulaire et de la séquence des requêtes effectuées.

### 3.8.2 Apprentissage par l'environnement

L'apprentissage par l'environnement consiste à récupérer des informations depuis le système d'indexation propre à AMI. Il s'agit d'un apprentissage par anticipation qui est basé sur l'idée relativement simple d'utiliser les connaissances acquises depuis la masse documentaire pour assister l'analyse.

Cette anticipation peut bien sûr influencer de manière notable l'interprétation d'une requête, voire même venir en opposition avec les éléments appris via l'utilisation du système. Mais, au final, c'est quand même le système documentaire qui fournit les données en réponse. Le positionnement du système est de "faire de son mieux" pour répondre aux questions en fonction des données disponibles et non d'établir la meilleure analyse de la demande pour conduire à un état d'échec par absence de résultats.

---

<sup>22</sup>Il faut rappeler que le système de recherche AMI fonctionne avec un lexique lié à son expérience mais avec aucun dictionnaire ni thésaurus.

<sup>23</sup>Par exemple, les cookies sur un service HTTP. Il est à noter que la fiabilité de cet identifiant est un gage des performances du système de *profiling*. Un cookie par exemple, qui est lié à une machine et non à un utilisateur humain, est un identifiant non fiable qui peut leurrer le système en associant des comportements humains divergents.

## A Modèles vectoriels

### A.1 Modèle vectoriel classique

Les modèles vectoriels sont des modèles d'indexation créés pour compenser la limitation de la pondération binaire des modèles booléens. Ils sont basés sur l'attribution de poids non binaires aux termes indexés dans les documents et aux termes des requêtes. Ces pondérations sont ensuite utilisées pour calculer le degré de similarité (*degree of similarity*) entre les documents du système et la requête de recherche. L'ordonnement par ordre décroissant des degrés de similarité permet d'avoir une estimation plus fiable de la pertinence des documents retrouvés [BYRN99, p.27].

Formellement, si  $T$  est le nombre de termes indexés (i.e. le nombre de termes trouvés dans les documents) et  $D$  le nombre de documents du système, on attribue un poids  $w_{i,j}$  au terme  $\tau_i$  contenu dans le document  $\delta_j$ . Un document est alors représenté comme un vecteur dans un espace de dimension  $T$  :

$$\vec{\delta}_j = (w_{1,j}, \dots, w_{T,j})$$

et une requête comme un autre vecteur dans ce même espace :

$$\vec{q}_j = (w_{1,q}, \dots, w_{T,q})$$

La notion de similarité est définie par le cosinus de l'angle entre deux vecteurs. Cette définition est valide pour la similarité entre deux documents ou entre une requête et un document :

$$\text{sim}(\delta_j, q) = \frac{\vec{\delta}_j \cdot \vec{q}}{|\vec{\delta}_j| \times |\vec{q}|}$$

soit :

$$\text{sim}(\delta_j, q) = \frac{\sum_{i=1}^T w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^T w_{i,j}^2} \times \sqrt{\sum_{i=1}^T w_{i,q}^2}}$$

Il reste à définir quels poids associer aux termes des documents. L'une des stratégies les plus répandues est la stratégie nommée *tf-idf* (*term frequency - inverse document frequency*) qui se base sur la fréquence  $\phi_{i,j}$  du terme  $\tau_i$  dans la document  $\delta_j$  et la fréquence normalisée  $f_{i,j}$  calculée par :

$$f_{i,j} = \frac{\phi_{i,j}}{\max_{\delta_i}(\phi_{i,j})}$$

Le poids  $w_{i,j}$  est alors calculé par :

$$w_{i,j} = f_{i,j} \times \log \frac{D}{n_i}$$

où  $n_i$  est le nombre de documents dans lequel le terme  $\tau_i$  apparaît. La seconde partie de l'équation représente la *fréquence inverse du document* (*inverse document frequency*) du terme  $\tau_i$ .

## A.2 Latent Semantic Indexing

L'idée principale du processus LSI est de projeter chaque document et chaque vecteur de requête dans un espace de dimension réduit (par rapport à la dimension initiale du système documentaire qui peut être très importante). Cet espace serait alors en association avec des "concepts". Le modèle est en fait basé sur la théorie des décompositions des valeurs singulières.

Le principe du LSI est de prendre la matrice  $M$ , de dimension  $T \times D$ , composée des poids  $w_{i,j}$  des termes  $\tau_i$  dans les documents  $\delta_j$  (§A.1 – p.21). Les poids sont, par exemple, ceux calculés par la méthode *tf-idf*.

Cette matrice  $M$  est décomposée, en utilisant la décomposition en valeurs singulières, en un produit de trois matrices :

$$M = K.S.\tilde{D}$$

où  $K$  est la matrice des valeurs propres de  $M$ ,  $\tilde{M}$ ,  $\tilde{D}$  est la matrice transposée de la matrice des valeurs propres de  $\tilde{M}.M$ , et  $S$  la matrice diagonale des valeurs singulières de dimension  $\sigma \times \sigma$  avec  $\sigma = \min(T, D)$ .

On conserve ensuite une sous-matrice de  $S$ , notée  $S_s$ , qui ne contient que les  $s$  ( $s < \sigma$ ) plus grandes valeurs singulières. On essaye d'avoir  $s \ll T$  et  $s \ll D$ , la difficulté du choix étant d'avoir  $s$  suffisamment grand pour que la sous-matrice soit représentative de la masse documentaire et  $s$  suffisamment petit pour filtrer les détails non pertinents.

Les autres matrices  $K$  et  $D$  sont alors réduites selon les lignes et colonnes conservées dans la matrice  $S$  afin de produire les sous-matrices  $K_s$  et  $D_s$ . On reconstitue ensuite la sous-matrice documentaire  $M_s$  telle que :

$$M_s = K_s.S_s.\tilde{D}_s$$

$M_s$  est la matrice la plus proche de  $M$  au sens de l'approximation des moindres carrés.

La relation entre deux documents est alors donnée par :

$$M_s \tilde{M}_s = (D_s S_s) (\tilde{D}_s \tilde{S}_s)$$

La méthode LSI introduit une conceptualisation intéressante du problème de recherche d'informations. Néanmoins, le cadre d'utilisation de la méthode est fortement dépendant de la collection de documents indexés. Elle obtient de bons résultats sur des masses documentaires très hétérogènes, où elle permet de dégager la sémantique sous-jacente de documents apparemment différents. Par contre, elle devient inefficace (voire même donne de moins bons résultats) lorsque les documents sont plus homogènes au niveau de leur vocabulaire. [MS99, p.564] considère également que l'utilisation d'une loi de distribution de Poisson ou binomiale négative serait plus judicieuse que la loi de distribution normale qui est implicitement la loi statistique utilisée dans la méthode.

## B Probabilités et réseaux bayésiens

### B.1 Origines

Le calcul des probabilités, dont l'origine vient des jeux de hasard<sup>24</sup>, est considérée comme une discipline scientifique à partir de 1654 par l'entremise de la correspondance entre Blaise Pascal et Pierre de Fermat sur deux problèmes posés par le Chevalier de Méré. Complété par la suite par les travaux de Huygens, de Bernoulli, de De Moivre et de Laplace, le calcul des probabilités ne devient réellement une discipline mathématique qu'après les travaux de Kolmogorov (1930) et le développement de la théorie de la mesure [Boc95].

### B.2 Probabilité conditionnelle et formule de Bayes

L'utilisation la plus fréquente des probabilités dans le domaine de la recherche d'informations est basée sur la notion de *probabilité conditionnelle*. Si  $(\Omega, \mathcal{A}, P)$  est un espace probabilisé et  $B$  un événement de cet espace avec une probabilité non nulle (i.e.  $P(B) \neq 0$ ), on définit la "probabilité conditionnelle de  $A$  sachant que  $B$  s'est réalisé" la probabilité :

$$P_B(A) = \frac{P(A \cap B)}{P(B)}$$

définie sur l'espace probabilisable  $(\Omega \cap B, \mathcal{A} \cap B)$ .

Si nous considérons maintenant une famille finie ou infinie dénombrable d'événements  $\{H_k, k=1,2,\dots\}$  deux à deux disjoints telle que  $\Omega = \bigcup_k H_k$ , alors pour tout événement  $A$ , on a la relation :

$$P(H_k|A) = \frac{P(H_k)P(A|H_k)}{\sum_k P(H_k)P(A|H_k)}$$

Il s'agit de la *formule de Bayes* qu'il est possible d'interpréter de la façon suivante : la famille  $\{H_k, k=1,2,\dots\}$  d'événements étant un recouvrement de  $\Omega$ , on considère la sous-famille  $\{H_{k_1}, H_{k_2}, \dots\}$  des événements dont l'intersection avec  $A$  est non vide ( $P(A \cap H_{k_i}) \neq 0$  et les autres événements ne peuvent pas se produire en même temps que  $A$  car la probabilité de l'intersection est nulle); si l'événement  $A$  se produit, c'est que l'un des événements de la sous famille s'est également produit. Il est alors possible de considérer les  $H_{k_i}$  comme les différentes *causes* possibles de  $A$  et aussi que la réalisation de l'événement  $A$  est un argument en faveur d'une des *hypothèses*  $H_{k_i}$ <sup>25</sup>.

<sup>24</sup>Le terme *hasard* désigne au XIIème siècle un jeu de dés. Ce terme vient de l'arabe *âz-âhr* qui signifie *les dés*. Quant à l'adjectif *aléatoire*, il vient du latin *alea* qui signifie également *les dés*.

<sup>25</sup>Cette interprétation est délicate à manipuler et peut conduire à des raisonnements logiques douteux comme le "problème du tricheur" proposé par Poincaré ou le calcul de la *loi de succession de Laplace* qui permet d'estimer qu'il y a une chance inférieure à  $10^{-6}$  que le soleil ne se lève pas demain matin.

### B.3 Réseaux bayésiens

Un réseau bayésien (*Bayesian network*) est un graphe où les nœuds sont des faits associés à des variables aléatoires et dont les arcs sont des relations de causalité, l'arc  $A \rightarrow B$  étant pondéré par la probabilité conditionnelle  $P(B|A)$ . Un réseau bayésien est en règle générale un graphe orienté acyclique afin d'éviter des boucles infinies lors du calcul d'inférence [Gac97].

L'inférence à partir d'un réseau bayésien consiste, lorsqu'une nouvelle information arrive en un nœud du graphe, à recalculer par propagation les probabilités marginales des arcs tout en respectant l'axiomatique des probabilités.

Il est à noter que l'approche des réseaux bayésiens est essentiellement différente (mais non antinomyque) de celle de la logique floue : les réseaux bayésiens s'attachent à la prise en compte de faits *précis, mais incertains*, alors que la logique floue s'intéresse à la modélisation de faits *imprécis* [BN99]

## C Réseaux neuronaux artificiels

### C.1 Historique

L'étude du calcul basé sur la simulation de l'activité cérébrale remonte aux travaux de McCulloch et Pitts (1943), suivi par la suite des travaux de Hebb (1949), notamment son célèbre "*Organization of behavior*". Les travaux préliminaires concernant l'intelligence artificielle ont conduit la communauté scientifique à se diviser en deux camps : ceux qui pensaient que les systèmes intelligents seraient construits sur des ordinateurs calquant le modèle cérébral et ceux, comme Minsky et Papert (1969), qui pensaient que l'intelligence proviendrait du calcul symbolique issu du modèle de la machine de von Neumann [Fu195].

L'approche du calcul symbolique a eu la faveur des chercheurs jusque dans les années 1970, les années 1980 voyant réapparaître un regain d'intérêt dans l'approche du calcul neuronal avec notamment :

- Hopfield qui a défini les fondations mathématiques pour la compréhension de la dynamique de plusieurs classes de réseaux ;
- Kohonen qui a développé l'apprentissage neuronal non supervisé dans des réseaux réguliers de neurones ;
- Rumelhart et McClelland qui ont introduit la notion de rétro-propagation (*backpropagation*) pour l'apprentissage dans le cas de réseaux multi-couches complexes.

Depuis, de nombreuses applications ont été réalisées sur le principe des réseaux neuronaux, les réseaux importants devant faire appel à des processeurs (chips en silicium) spécialisés.

### C.2 Modélisation

Un neurone humain est constitué (d'une manière très schématique dans notre description) de plusieurs protusions appelées dendrites et d'une longue branche appelée axone qui le connecte à d'autres neurones via les synapses.

Un neurone reçoit des impulsions électriques via ses synapses et, si la somme des impulsions dépasse un seuil fixé, le neurone transmet une information aux neurones qui lui sont connectés via son axone. Les synapses peuvent être de nature excitatoire ou inhibitrice selon que les impulsions parvenant au neurone sont ajoutées ou retranchées de la somme des impulsions [Ros95].

Formalisé mathématiquement, un neurone artificiel est perçu comme un système recevant plusieurs impulsions en entrée et générant, en cas de dépassement d'un seuil, une impulsion en sortie :

$$y = F\left(\sum_i w_i x_i - \theta\right)$$

Les variables  $x_i$  représentent les impulsions reçues par le neurone depuis les autres neurones, les variables  $w_i$  sont les importances relatives des connexions

synaptiques (excitatoire si  $w_i > 0$  ou inhibitrice si  $w_i < 0$ ), la fonction  $F$  étant une fonction non linéaire qui définit le type de signal restitué par le neurone. Habituellement,  $F$  est une fonction sigmoïdale ( $1 + \frac{1}{e^{-s}}$ ) (*sigmoid function*) ou une fonction en escalier (*step function*).

### C.3 Apprentissage et rétro-propagation

Un réseau de neurones requiert en fait un apprentissage initial afin de parvenir ensuite à extraire, à partir des spécificités du jeu d'apprentissage, des généralités dans le cadre d'une utilisation postérieure.

L'apprentissage s'effectue au moyen d'un algorithme appelé rétro-propagation (*backpropagation*) qui, en plaçant en entrée du réseau des impulsions dont on connaît le résultat à l'avance, permet d'ajuster les poids synaptiques  $w_i$  en minimisant l'erreur quadratique calculée à partir des différences entre les résultats observés et les résultats désirés.

## D Logique floue

### D.1 Présentation

#### D.1.1 Historique

La logique floue (*fuzzy logic*) a été formalisée en 1965 par Lotfi Zadeh, professeur émérite à l'Université de Berkeley (Californie). Afin de poursuivre ses travaux en automatique et en théorie des systèmes, Lotfi Zadeh a éprouvé le besoin de formaliser la représentation et le traitement de connaissances imprécises ou approximatives, afin de pouvoir traiter des systèmes de grande complexité dans lesquels interviennent des facteurs humains [BM95, p.2].

Il est à noter que la logique floue est une extension de la logique classique (booléenne) et que les deux théories sont identiques et parviennent aux mêmes résultats lorsque les conditions limites (toute valeur est *vraie* ou *fausse*) de la logique classique sont atteintes. De la même manière, la logique utilisée dans la logique floue *n'est pas* floue : la logique est un édifice mathématique totalement cohérent utilisant les mêmes règles que les mathématiques usuelles, seule l'axiomatique de base est différente.

La logique floue a reçu un très mauvais accueil aux États-Unis lors de la publication des travaux de Zadeh et elle reste encore entachée de cette mauvaise réputation. Les universitaires américains ont depuis largement changé d'attitude, surtout depuis les succès commerciaux des pays asiatiques qui ont, eux, largement adopté cette théorie <sup>26</sup>.

L'un des résultats les plus troublants de l'approche de la logique floue est l'invalidation du *principe de non-contradiction* (*law of contradiction*), qui spécifie qu'une chose et son contraire ne peuvent être vrais simultanément, et du *principe du tiers exclus* (*law of the excluded middle*), qui indique qu'au moins une chose ou son contraire doivent être vrais. Les règles classiques de raisonnement, basées sur le *modus ponens* et le *modus tolens*, sont également invalides, ce qui conduit à redéfinir les règles de la logique des prédicats.

#### D.1.2 Imprécision et incertitude

Les premiers travaux concernant la logique floue ont concerné les données imprécises. La modélisation de l'imprécision se faisant au moyen de la théorie des sous-ensembles flous. La notion d'incertitude a été formalisée plus tardivement par la théorie des possibilités (§E – p.32).

Il est bien sûr possible de croiser les théories pour obtenir :

---

<sup>26</sup>Le terme *fuzzy* a été déclaré mot des années 90 sur le plan technologique au Japon. Ses domaines d'applications dans le domaine de l'automatisme sont présents dans les objets les plus usuels : systèmes de climatisation, appareils photographiques et caméscopes auto-focus, lave-linges, etc. Le métro automatique de Saipan (Japon) est entièrement contrôlé par un système à base de logique floue.

- des raisonnements certains sur des données imprécises ;
- des raisonnements incertains sur des données précises ;
- des raisonnements incertains sur des données imprécises.

Il est important de noter que la logique floue ne vient pas concurrencer d'autres techniques, mais vient en complément de formalisation lorsqu'on doit faire face à l'imprécision ou l'incertitude. On peut ainsi trouver :

- des réseaux de neurones utilisant la logique floue : les réseaux neuro-flous (*fuzzy neuro-networks*) ;
- la logique floue pour modéliser les données imprécises dans le cadre de l'inférence bayésienne (*fuzzy bayesian inference*) ;
- l'imprécision de mesures probabilistes (*fuzzy probabilities*).

## D.2 Définitions

Soit  $X$  un ensemble de référence (fini ou non). On note  $\mathcal{P}(X)$  l'ensemble des parties de  $X$ , c'est-à-dire l'ensemble de tous les sous-ensembles (au sens strict de la théorie des ensembles) de  $X$  :

$$A \in \mathcal{P}(X) \iff A \subseteq X$$

Un sous-ensemble strict (ou *classique*) est défini par une fonction caractéristique  $\chi_A$  telle que :

$$\begin{cases} \chi_A : X \longrightarrow \{ 0, 1 \} \\ \chi_A(a) = 1 \text{ si } a \in A \\ \chi_A(a) = 0 \text{ si } a \notin A \end{cases}$$

Un sous-ensemble flou (*fuzzy set*)<sup>27</sup>  $A$  de  $X$  est défini par une fonction d'appartenance  $\Phi_A$  qui associe, à chaque élément de  $X$ , le degré d'appartenance de cet élément à  $A$  :

$$\Phi_A : X \longrightarrow [ 0, 1 ]$$

Si la fonction d'appartenance ne prend que les valeurs 0 et 1, le sous-ensemble flou devient un sous-ensemble classique (qui n'est donc bien qu'un cas particulier).

On notera  $\mathcal{F}(X)$  l'ensemble des sous-ensembles flous de  $X$ .

## D.3 Caractéristiques d'un sous-ensemble flou

Un sous-ensemble flou possède des caractéristiques qui permettent de définir en quoi il diffère d'un sous-ensemble classique. La première de ces caractéristiques est le *support* de  $A$  :

$$\text{supp}(A) = \{ x \in X \mid \Phi_A(x) \neq 0 \}$$

<sup>27</sup>La dénomination *ensemble flou* est un abus de langage découlant directement de la traduction de *fuzzy set*. La définition de la notion de *flou* s'effectue sur des sous-ensembles d'un ensemble.

Le support de  $A$  est un sous-ensemble classique qui identifie les éléments qui appartiennent au moins un peu à  $A$ .

Une deuxième caractéristique d'un sous-ensemble flou est sa *hauteur* notée  $h(A)$  :

$$h(A) = \sup_{x \in X} (\Phi_A(x))$$

qui représente la plus grande valeur d'appartenance d'un élément de  $X$  à  $A$ . Une famille importante de sous-ensembles flous sont ceux qui sont *normalisés*, c'est-à-dire ceux pour lesquels au moins un élément de  $X$  prend la valeur 1 (autrement dit si  $h(A) = 1$ ).

Une troisième caractéristique est le *noyau* de  $A$  :

$$\text{noy}(A) = \{ x \in X \mid \Phi_A(x) = 1 \}$$

qui définit les éléments qui appartiennent de façon absolue à  $A$ .

Pour un sous-ensemble strict, nous avons  $\text{supp}(A) = \text{noy}(A) = A$  et  $h(A) = 1$ , ce qui signifie qu'un sous-ensemble strict est normalisé et est toujours identique à son noyau et à son support.

Enfin, il est possible de définir la *cardinalité* d'un sous-ensemble flou d'un ensemble fini  $X$  :

$$|A| = \sum_{x \in X} \Phi_A(x)$$

ce qui, dans le cas d'un sous-ensemble strict, correspond bien au nombre d'éléments appartenant à  $A$ .

#### D.4 Opérations sur les sous-ensembles flous

Afin de pouvoir manipuler les sous-ensembles flous, il est nécessaire de définir les notions d'inclusion, d'intersection, d'union et de complément qui sont connues de la théorie des ensembles et qui doivent être des extensions des opérations classiques, c'est-à-dire avoir le même comportement lorsque les sous-ensembles flous deviennent des sous-ensembles stricts.

On définit d'abord l'*égalité* de deux sous-ensembles flous :

$$\forall A, B \in \mathcal{F}(X) \quad A = B \iff \forall x \in X \quad \Phi_A(x) = \Phi_B(x)$$

qui correspond à l'identité des fonctions d'appartenance des deux sous-ensembles flous.

L'*appartenance* d'un sous-ensemble flou à un autre sous-ensemble flou est défini par :

$$\forall A, B \in \mathcal{F}(X) \quad A \subseteq B \iff \forall x \in X \quad \Phi_A(x) \leq \Phi_B(x)$$

qui correspond à l'idée que  $A$  est inclus dans  $B$  si tout élément de  $A$  appartient aussi à  $B$  avec un degré d'appartenance au moins aussi élevé.

L'*intersection* de deux sous-ensembles flous est définie par une fonction d'appartenance qui correspond à la plus petite valeur d'appartenance à l'un des sous-ensembles flous :

$$\forall x \in X \quad \Phi_{A \cap B}(x) = \min(\Phi_A(x), \Phi_B(x))$$

un élément ne pouvant en effet appartenir à  $A$  et à  $B$ , donc à  $A \cap B$ , que moins fortement qu'il n'appartient à chacun des sous-ensembles flous.

De manière identique, l'*union* de deux sous-ensembles flous est définie par une fonction d'appartenance qui correspond à la plus grande valeur d'appartenance à l'un des sous-ensembles flous :

$$\forall x \in X \quad \Phi_{A \cup B}(x) = \max(\Phi_A(x), \Phi_B(x))$$

un élément devant appartenir à  $A \cup B$  d'une manière au moins aussi forte qu'à chacun des sous-ensembles flous.

Il est possible de montrer que ces opérations respectent les propriétés définies pour les sous-ensembles stricts dans la théorie des ensembles, à savoir :

- Les opérateurs  $\cap$  et  $\cup$  sont associatifs.
- L'opérateur  $\cup$  est commutatif.
- L'opérateur  $\cap$  est commutatif.
- $\forall A \in \mathcal{F}(X) \quad A \cup \emptyset = A, \quad A \cup X = X$
- $\forall A \in \mathcal{F}(X) \quad A \cap \emptyset = \emptyset, \quad A \cap X = A$
- $\forall A, B \in \mathcal{F}(X) \quad A \cap B \subseteq A \subseteq A \cup B$
- Les opérateurs  $\cap$  et  $\cup$  sont distributifs l'un par rapport à l'autre.
- $|A| + |B| = |A \cup B| + |A \cap B|$

Le *complément* d'un sous-ensemble flou est défini par la fonction d'appartenance :

$$\forall x \in X \quad \Phi_{\bar{A}}(x) = 1 - \Phi_A(x)$$

qui indique que moins un élément appartient à  $A$  plus il appartient au complémentaire  $\bar{A}$  de  $A$ .

Les propriétés suivantes sont identiques à celles de la théorie des ensembles :

- $\forall A, B \in \mathcal{F}(X) \quad \overline{(A \cap B)} = \bar{A} \cup \bar{B}$
- $\forall A, B \in \mathcal{F}(X) \quad \overline{(A \cup B)} = \bar{A} \cap \bar{B}$
- $\forall A \in \mathcal{F}(X) \quad \overline{(\bar{A})} = A$
- $\forall A \in \mathcal{F}(X) \quad |A| + |\bar{A}| = |X|$

Par contre, contrairement à cette même théorie, nous n'avons pas le respect de la règle  $A \cap \bar{A} = \emptyset$  (ce qui invalide le principe logique de *non-contradiction*, à savoir qu'un élément peut appartenir à un sous-ensemble flou et à son complémentaire), ainsi que de la règle  $A \cup \bar{A} = X$  (ce qui invalide le principe logique du *tiers-exclus* qui impose qu'un élément appartienne au moins à un sous-ensemble ou à son complémentaire).<sup>28</sup>

<sup>28</sup> Le choix des fonctions min, max et du complément à 1 pour définir les opérations  $\cap$ ,  $\cup$  et

### D.5 $\alpha$ -coupes associées à un sous-ensemble flou

La notion d' $\alpha$ -coupe permet d'obtenir, à partir d'ensembles flous, des ensembles stricts sur lesquels on peut appliquer les règles habituelles de la théorie des ensembles. L'idée est définir un seuil  $\alpha$  ( $\alpha \in [0, 1]$ ) servant de limite inférieure pour l'appartenance des éléments.

On définit ainsi l'ensemble strict  $A_\alpha$  de  $\mathcal{P}(X)$  associé au sous-ensemble flou  $A$  de  $\mathcal{F}(X)$  par :

$$A_\alpha = \{x \in X \mid \Phi_A(x) \geq \alpha\}$$

Les  $\alpha$ -coupes de  $A$  sont des sous-ensembles non flous de  $X$  emboîtées par rapport à la valeur de  $\alpha$ , c'est-à-dire que  $A_\alpha \subseteq A_{\alpha'}$  si  $\alpha \leq \alpha'$ .

La restriction d'un ensemble flou à une  $\alpha$ -coupe est une opération compatible avec les opérations classiques sur les ensembles :

- $\forall A, B \in \mathcal{F}(X) \quad (A \cap B)_\alpha = A_\alpha \cap B_\alpha$
- $\forall A, B \in \mathcal{F}(X) \quad (A \cup B)_\alpha = A_\alpha \cup B_\alpha$
- $\forall A, B \in \mathcal{F}(X) \quad A \subseteq B \Rightarrow A_\alpha \subseteq B_\alpha$

Notamment, si  $\alpha = 0$  alors  $A_\alpha = X$ , et si  $\alpha = 1$  alors  $A_\alpha = \text{noy}(A)$ .

### D.6 Produit cartésien de sous-ensembles flous

Soit  $X_1, X_2, \dots, X_p$  des ensembles de référence et  $X$  leur produit cartésien  $X = X_1 \times X_2 \times \dots \times X_p$  dont les éléments sont des  $p$ -uplets  $(x_1, x_2, \dots, x_p)$  avec  $x_i \in X_i$ .

A partir des sous-ensembles flous  $A_{i,i=1,\dots,p}$  définis respectivement sur  $X_{i,i=1,\dots,p}$ , on construit un sous-ensemble flou  $A = A_1 \times A_2 \times \dots \times A_p$  de  $X$  considéré comme le produit cartésien des  $A_i$  ayant comme fonction d'appartenance :

$$\forall x = (x_1, x_2, \dots, x_p) \in X, \quad \Phi_A(x) = \min(\Phi_{A_1}(x_1), \dots, \Phi_{A_p}(x_p))$$

### D.7 Relations et quantités floues

Soit  $X$  et  $Y$  deux ensembles, une *relation floue*  $\mathcal{R}$  entre  $X$  et  $Y$  est définie comme un sous-ensemble flou de  $X \times Y$ . Si  $X$  et  $Y$  sont deux ensembles finis, la relation floue  $\mathcal{R}$  peut être complètement définie par la matrice des valeurs de sa fonction d'appartenance.

---

la complémentarité est justifié par le fait que ce sont ces opérateurs qui préservent au mieux la structure de la théorie des ensembles sans toutefois pouvoir préserver ces deux dernières règles. D'autres opérateurs sont envisageables (normes et co-normes triangulaires, négation pour le complément) mais, même s'ils autorisent de conserver ces deux règles, ils ne permettent pas de conserver autant de propriétés et sont donc, globalement, moins optimaux.

## E Théorie des possibilités

### E.1 Présentation

La théorie des possibilités a été introduite en 1978 par Lotfi Zadeh afin de prendre en compte la représentation de la notion d'incertitude sur la véracité d'une affirmation, que les données manipulées dans cette affirmation soient précises ou non. Si la théorie des sous-ensembles flous permet de traiter les termes "à peu près" ou "environ", la théorie des possibilités intègre, quant à elle, la notion de variables linguistiques qui gèrent des notions comme "très", "beaucoup", "peu" ou "plutôt".

La théorie des possibilités définit les notions de *mesures de possibilité* et de *mesures de nécessité* qui permettent de raisonner via la *logique possibiliste*. On peut néanmoins montrer que cette théorie est un cas particulier d'un formalisme plus général, appelé *théorie de l'évidence*, qui, via la notion de *mesure floue*, est une base commune à la théorie des possibilités et à la théorie des probabilités [BM95, p.84].

Ces deux dernières théories sont donc "cousines" mais divergent fondamentalement sur les contraintes imposées aux éléments focaux (un élément focal étant une généralisation de la notion d'événement en probabilité). La théorie des probabilités requérant une meilleure connaissance des événements possibles que la théorie des possibilités <sup>29</sup>.

Nous pouvons considérer que la théorie des possibilités est en fait un substitut possible à la théorie des probabilités, cette dernière ayant fait l'objet de plus de travaux par le fait même qu'elle est plus ancienne sur le plan théorique.

### E.2 Mesure de possibilité

Une *mesure de possibilité* associe à tout sous-ensemble strict d'un ensemble de référence  $X$  un coefficient compris entre 0 et 1. La mesure de possibilité  $\Pi$  est définie par :

$$\begin{cases} \Pi(\emptyset) = 0 \\ \Pi(X) = 1 \\ \forall A_i, i \in 1, \dots, n \in \mathcal{P}(X) \quad \Pi(\bigcup_{i \in 1, \dots, n} A_i) = \sup_{i \in 1, \dots, n} \Pi(A_i) \end{cases}$$

Dans le cas de deux sous-ensembles, la dernière relation se réduit à :

$$\forall A, B \in \mathcal{P}(X)^2 \quad \Pi(A \cup B) = \max(\Pi(A), \Pi(B))$$

<sup>29</sup>La notion d'événements indépendants, qui autorise en théorie des probabilités l'addition des probabilités, n'est par exemple plus satisfaite en théorie des possibilités. Il s'agit d'une des propriétés fondamentales en probabilité.

ce qui signifie que la réalisation de l'un des deux événements  $A$  ou  $B$ , pris indifféremment, est affectée du même coefficient de possibilité que la réalisation de l'événement le plus probable. Un événement  $A$  est tout à fait possible si  $\Pi(A) = 1$ , et impossible si  $\Pi(A) = 0$ .

Par contre, le coefficient attribué à l'intersection de parties de  $X$  n'est pas défini. La seule relation imposée est que le coefficient de l'intersection soit majoré par le plus petit des coefficients attribués aux événements :

$$\forall A, B \in \mathcal{P}(X)^2 \quad \Pi(A \cap B) \leq \min(\Pi(A), \Pi(B))$$

On a en effet :  $\Pi(A) = \Pi((A \cap B) \cup A) = \max(\Pi(A \cap B), \Pi(A)) \geq \Pi(A \cap B)$

De même :  $\Pi(B) = \Pi((A \cap B) \cup B) = \max(\Pi(A \cap B), \Pi(B)) \geq \Pi(A \cap B)$

D'où la relation puisque  $\Pi(A \cap B)$  est à la fois plus petit que  $\Pi(A)$  et  $\Pi(B)$ .

On en déduit que deux événements peuvent être possibles (c'est-à-dire  $\Pi(A) \neq 0$  et  $\Pi(B) \neq 0$ ) mais que leur occurrence simultanée soit impossible ( $\Pi(A \cap B) = 0$ ).

On en déduit également qu'une mesure de possibilité est monotone relativement à la relation d'ordre partiel d'inclusion définie sur  $\mathcal{P}(X)$  :

$$\forall A, B \in \mathcal{P}(X)^2 \quad A \subseteq B \Rightarrow \Pi(A) \leq \Pi(B)$$

En effet, si  $A \subseteq B$ ,  $A = A \cap B$  et  $\Pi(A) = \Pi(A \cap B) \leq \min(\Pi(A), \Pi(B)) \leq \Pi(B)$ .

Prenons maintenant une partie  $A$  de  $X$  et son contraire (i.e.  $\bar{A}$ ). Au moins l'un des deux événements est tout à fait possible<sup>30</sup>, ce qui nous donne les relations suivantes :

$$\forall A \in \mathcal{P}(X) \quad \max(\Pi(A), \Pi(\bar{A})) = 1$$

$$\forall A \in \mathcal{P}(X) \quad \Pi(A) + \Pi(\bar{A}) \geq 1$$

### E.3 Mesure de nécessité

Une mesure de possibilité fournit une information sur l'occurrence d'un événement  $A$  mais ne fournit aucune information pour décrire l'incertitude existante concernant  $A$ . Notamment, on peut avoir  $\Pi(A) = 1$ , ce qui signifie que l'événement est tout à fait réalisable, mais il est aussi possible d'avoir  $\Pi(\bar{A}) = 1$ , auquel cas on a une indétermination complète sur la réalisation de  $A$ , ou  $\Pi(\bar{A}) = 0$ , auquel cas seul  $A$  peut être réalisé, signifiant ainsi que  $A$  est un événement certain.

Afin de compléter l'information sur  $A$ , on cherche à indiquer le degré avec lequel la réalisation de  $A$  est certaine au moyen d'une *mesure de nécessité*. Cette mesure constitue une grandeur duale de la mesure de possibilité.

<sup>30</sup>Nous raisonnons ici sur des sous-ensembles stricts pour lesquels le principe de non-contradiction s'applique.

La mesure de nécessité est une fonction de  $\mathcal{P}(X)$  dans  $[0, 1]$  telle que :

$$\begin{cases} N(\emptyset) = 0 \\ N(X) = 1 \\ \forall A_i, i \in \{1, \dots, n\} \in \mathcal{P}(X) \quad N(\bigcap_{i \in \{1, \dots, n\}} A_i) = \inf_{i \in \{1, \dots, n\}} N(A_i) \end{cases}$$

Dans le cas de deux sous-ensembles, la dernière relation se réduit à :

$$\forall A, B \in \mathcal{P}(X)^2 \quad N(A \cap B) = \min(N(A), N(B))$$

Cette relation impose la monotonie de la mesure de nécessité par rapport à la relation d'ordre partiel définie sur  $\mathcal{P}(X)$  par l'inclusion :

$$\forall A, B \in \mathcal{P}(X)^2 \quad A \subseteq B \Rightarrow N(A) \leq N(B)$$

En effet, si  $A \subseteq B$ ,  $A = A \cap B$  et  $N(A) = N(A \cap B) = \min(N(A), N(B)) \leq N(B)$ .

La mesure de nécessité de l'union de deux événements n'est pas définie. Par contre, en vertu de la monotonie précitée, elle vérifie la relation :

$$\forall A, B \in \mathcal{P}(X)^2 \quad N(A \cup B) \geq \max(N(A), N(B))$$

On a en effet :  $N(A \cup B) \geq N(A)$  car  $A \subseteq A \cup B$

De même :  $N(A \cup B) \geq N(B)$  car  $B \subseteq A \cup B$

D'où la relation puisque  $N(A \cup B)$  est à la fois plus grand que  $N(A)$  et  $N(B)$ .

On en déduit également les relations entre les mesures de nécessité d'un événement et de son contraire :

$$\forall A \in \mathcal{P}(X) \quad \min(N(A), N(\bar{A})) = 0$$

$$\forall A \in \mathcal{P}(X) \quad N(A) + N(\bar{A}) \leq 1$$

## Références

- [Abd94] Hervé Abdi. *Les réseaux de neurones*. Presses Universitaires de Grenoble, 1994.
- [BB97] George Bojadziev and Maria Bojadziev. *Fuzzy logic for business, finance, and management*. World Scientific Publishing Co. Pte. Ltd., 1997.
- [BM94] Bernadette Bouchon-Meunier. *La logique floue*. Presses Universitaires de France, 1994.
- [BM95] Bernadette Bouchon-Meunier. *La logique floue et ses applications*. Addison-Wesley France, 1995.
- [BN99] Ann Becker and Patrick Naïm. *Les réseaux bayésiens*. Eyrolles, 1999.
- [Boc95] Nino Boccara. *Probabilités*. Ellipses, 1995.
- [Bou98] Pierrette Bouillon. *Traitement automatique des langues naturelles*. Aupelf-Uref-Editions Duculot, 1998.
- [BYN] Ricardo Baeza-Yates and Gonzalo Navarro. *Fast Approximate String Matching in a Dictionary*. Dept. of Computer Science, University of Chile.
- [BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
- [CHL01] Maxime Crochemore, Christophe Hancart, and Thierry Lecroq. *Algorithmique du texte*. Vuibert, 2001.
- [CHT99] Nick Craswell, David Hawking, and Paul B. Thistlewaite. Merging results from isolated search engines. In *Australasian Database Conference*, pages 189–200, 1999.
- [Fu95] Robert Fullér. *Neural Fuzzy Systems*. Åbo Akademi University, 1995.
- [Gac97] Louis Gacôgne. *Éléments de logique floue*. Hermès, 1997.
- [Ja00] Christian Jacquemin and al. *Traitement automatique des langues pour la recherche d'informations – Volume 41*. ATALA/Hermès Science Publications, 2000.
- [JM00] Daniel Jurafsky and James H. Martin. *Speech and language processing*. Prentice Hall, 2000.
- [Jon86] Karen Sparck Jones. *Synonymy and semantic classification*. Edinburgh Information Rechnology Series, 1986.
- [Mit97] Tom M. Mitchell. *Machine Learning*. McGraw-Hill International Editions, 1997.
- [MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. The MIT Press, 1999.
- [RHS03] Yves Rasolofo, David Hawking, and Jacques Savoy. Result merging strategies for a current news metasearcher. *Inf. Process. Manage.*, 39(4) :581–609, 2003.

- [Ros95] Timothy J. Ross. *Fuzzy logic with engineering applications*. International Edition, 1995.
- [Sch] Hinrich Schütze. *Dimensions of meaning*. Center for the study of language and information – Standaford, CA 94305-4115.
- [Sil93] Max Silberztein. *Dictionnaires électroniques et analyse automatique de textes*. Masson, 1993.
- [VGJL94] Ellen M. Voorhees, Narendra Kumar Gupta, and Ben Johnson-Laird. The collection fusion problem. In *Text REtrieval Conference*, pages 0–, 1994.
- [Wit68] Ludwig Wittgenstein. *Philosophical Investigations [Philosophische Untersuchungen]*. Oxford : Basil Blackwell, third edition, 1968.