RECITAL 2003, Batz-sur-Mer, 11–14 juin 2003

Société d'agents apprenants et sémantique lexicale : comment construire des vecteurs conceptuels à l'aide de la double boucle

Didier Schwab
LIRMM
Laboratoire d'informatique, de Robotique
et de Microélectronique de Montpellier
MONTPELLIER - FRANCE.
schwab@lirmm.fr
http://www.lirmm.fr/~schwab

Mots-clefs – Keywords

sociétés d'agents, vecteurs conceptuels, sémantique lexicale, apprentissage
agents society, conceptual vectors, lexical semantic, learning

Résumé - Abstract

Dans le cadre de la représentation du sens en TALN, nous développons actuellement un système d'analyse des aspects thématiques des textes et de désambiguïsation lexicale basée sur les vecteurs conceptuels. Ces vecteurs visent à représenter un ensemble d'idées associées à tout segment textuel. À partir de ce modèle, nous avons posé des hypothèses sur la construction des vecteurs. Dans cet article, nous montrons comment ces hypothèses, ainsi que des considérations techniques comme la possibilité de distribuer les tâches à effectuer ou la modularité, nous ont amenées à adopter une architecture multi-agents. Chaque agent possède un certain nombre de compétences, une mémoire qui lui est propre et peut interragir avec son environnement (les autres agents). Pour finir, nous présentons les agents déjà implémentés et un exemple de leur collaboration.

In the framework of research in meaning representations in NLP, we focus our attention on thematic aspects and lexical disambiguation based on conceptual vectors. These vectors are supposed to encode "ideas" associated to words or expressions. Starting from this model, we have built a number of hypothesis on the construction of vectors. In this article, we show how we adopted a multi-agents architecture using these hypothesis together with some technical considerations such as modularity and tasks distribution. Each agent has some abilities, a memory, and can interact with its environment (the other agents). To conclude, we present implemented agents and an example of their collaboration.

1 introduction

Dans le cadre de la représentation du sens en TALN, l'équipe TAL (Traitement Algorithmique des Langues) du LIRMM développe actuellement un système d'analyse des aspects thématiques des textes et de désambiguïsation lexicale basée sur les vecteurs conceptuels. Ces vecteurs cherchent à représenter un ensemble d'idées associées à tout segment textuel (mots, expressions, textes, ...). À partir de ce modèle, nous avons posé un certain nombre hypothèses sur la construction des vecteurs (génération automatique, multi-sources, apprentissage permanent, ...). Dans cet article, nous montrons comment ces hypothèses ainsi que des considérations techniques comme la possibilité de distribuer les tâches à effectuer ou la modularité, nous ont amené à constituer une architecture multi-agents. Chaque agent possède un certain nombre de compétences (conservations de définitions, de vecteurs, analyse morpho-syntaxiques, fonctions lexicales, ...), d'une mémoire qui lui est propre et peut interragir avec son environnement (les autres agents). Nous présentons l'organisation générale d'un tel système et enfin un exemple de coopération entre agents dans le cadre de l'apprentissage d'un item.

2 Vecteurs conceptuels

Nous représentons les aspects thématiques des segments textuels (documents, paragraphes, syntagmes, etc) par des vecteurs conceptuels. Les vecteurs ont été utilisés en informatique documentaire pour la recherche d'information (Salton et MacGill, 1983). Leur emploi pour la représentation du sens est plus le fait du modèle LSI (*Latent Semantic Indexing* (Deerwester et al., 90)) issue de l'analyse sémantique latente en psycho-linguistique. En informatique, et de façon presque concurrente, c'est à partir de (Chauché, 90) que l'on a une formalisation de la projection de la notion, linguistique cette fois, de champ sémantique dans un espace vectoriel. À partir d'un ensemble de notions élémentaires dont nous faisons l'hypothèse, les concepts, il est possible de construire des vecteurs (dits conceptuels) et de les associer à des items lexicaux[1]. Les termes polysémiques combinent les différents vecteurs correspondant aux différents sens. Cette approche vectorielle est fondée sur des propriétés mathématiques bien connues sur lesquelles il est possible d'effectuer des manipulations formellement pertinentes auxquelles sont attachées des interprétations linguistiques raisonnables. Les concepts sont donnés *a priori*. Dans notre expérimentation sur le français nous utilisons (Larousse, 1992) dans lequel sont définis 873 concepts. L'hypothèse principale du thésaurus, que nous adoptons ici, est que cet ensemble constitue un espace générateur pour les termes et leurs sens. D'une façon plus générale, n'importe quel sens peut s'y projeter selon le principe suivant.

Soit $\mathcal{C}$ un ensemble fini de n concepts, un vecteur conceptuel V est une combinaison linéaire d'éléments c_i de $\mathcal{C}$. Pour une idée A, le vecteur V_A est la description en extension des activations de tous les concepts de $\mathcal{C}$. Par exemple, les différents sens d'‹*existence*› peuvent être projetés sur les concepts suivants (les *CONCEPT*⌈*intensité*⌋ sont ordonnés par valeurs décroissantes) : V ‹*existence*› = (*EXISTENCE*⌈*0.82*⌋, *VIE*⌈*0.44*⌋, *IDENTITÉ*⌈*0.38*⌋, *ÉTAT*⌈*0.33*⌋, ...). En pratique, plus $\mathcal{C}$ est large, plus fines sont les descriptions de sens mais plus leur manipulation est lourde. Il est clair que pour les vecteurs denses, ceux qui ont peu de coordonnées nulles, l'énumération des concepts activés est longue et la pertinance difficile à évaluer. En général, pour évaluer la qualité

[1] Les items lexicaux sont des mots ou des expressions qui constituent les entrées du lexique. Par exemple, ‹*voiture*› ou ‹*pomme de terre*› sont des items lexicaux. Dans la suite, par abus de langage, nous utiliserons parfois mot ou terme pour qualifier un item lexical. Nous noterons les items en minuscule et entre apostrophes (‹*vie*›) et les concepts en majuscules (*VIE*).

d'un vecteur, nous préfèrerons sélectionner les termes thématiquements proches, le *voisinage* (noté $\mathcal{V}$). Par exemple, pour ‹*vie*› : $\mathcal{V}$(‹*existence*›) : ‹*existence*›, ‹*exister*›, ‹*vivant*›, ‹*vie*›, ... Cette opération est réalisée à l'aide de la distance angulaire.

2.1 Distance angulaire

Soit $Sim(X, Y)$ une des mesures de *similarité* entre deux vecteurs X et Y, souvent utilisée en recherche d'information (Morin, 1999). $Sim(X, Y) = \cos(\widehat{X, Y}) = \frac{X \cdot Y}{\|X\| \times \|Y\|}$ avec "$\cdot$" désignant le produit scalaire. Nous supposons ici que les composants des vecteurs sont positifs ou nuls, la *distance angulaire* entre deux vecteurs X et Y est $D_A(X, Y) = \arccos(Sim(X, Y))$. Intuitivement, cette fonction constitue une évaluation de la *proximité thématique* et en pratique la mesure de l'angle entre les deux vecteurs. Nous considérons en général que pour une distance $D_A(X, Y) \leq \frac{\pi}{4}$ (45°), X et Y sont thématiquement proches et partagent plusieurs concepts. Pour $D_A(X, Y) \geq \frac{\pi}{4}$, la proximité thématique est considérée comme faible et aux alentours de $\frac{\pi}{2}$ (90°), X et Y n'ont aucune relation. On remarquera que ces seuils ne servent que d'indicateurs pour un réviseur humain et restent à la fois subjectifs et arbitraires. D_A est une distance, elle vérifie donc les propriétés de réflexivité, de symétrie et d'inégalité triangulaire. Nous obtenons, par exemple, les angles suivants[2] :

D_A(V(‹*locomotive*›), V(‹*locomotive*›))=0 (0°)
D_A(V(‹*locomotive*›), V(‹*locomotrice*›))=0.24 (14°)
D_A(V(‹*locomotive*›), V(‹*automotrice*›))=0.22 (13°)
D_A(V(‹*locomotive*›), V(‹*rhododendron*›))=1.15 (65°)
D_A(V(‹*locomotive*›), V(‹*train*›))=0.54 (31°)
D_A(V(‹*locomotive*›), V(‹*guépard*›))=0,94 (54°)

Le premier résultat a une interprétation directe, ‹*locomotive*› ne peut être plus proche d'autre chose que de lui même. Les termes ‹*automotrice*› et ‹*locomotrice*› sont synonymes de ‹*locomotive*›, ce qui explique les deux résultats suivants. Le peu de rapport entre ‹*locomotive*› et ‹*rhododendron*› explique l'écart entre leur vecteurs. Dans le dernier exemple, l'angle peu important entre ‹*locomotive*› et ‹*guépard*› au regard de celui entre ‹*locomotive*› et ‹*rhododendron*› se comprend si on se rappelle que D_A est une distance thématique et non une distance ontologique. Les deux items ont en commun de partager une idée de rapidité. On remarquera que les comparaisons entre les valeurs sont plus significatives que les valeurs elles-mêmes. Seule une expertise humaine est capable de juger de la pertinence des vecteurs (si les résultats renvoyés sont cohérents avec la langue).

2.2 Construction des vecteurs conceptuels : hypothèses de départ

Tout en respectant le modèle présenté, les vecteurs peuvent être construits de plusieurs manières. Il s'agit d'une étape clé car les résultats pratiques, la proximité thématique en particulier, sont différents suivant le mode de construction. La construction de nos vecteurs est basée sur quelques hypothèses fortes.

2.2.1 Génération automatique

Le but est de construire une base de stockage de couples $< item, vecteur >$. La difficulté principale vient de l'affectation de vecteurs à chaque item. Dans notre expérience sur le français, pour un peu plus de 100000 entrées (mots communs, nom propres, expressions, ...), le taux de termes polysémiques est d'environ 61%. Le nombre moyen de définitions pour ces derniers étant d'un peu plus de 5, il faudrait indexer à la main plus de 400000 vecteurs de taille 873, ce qui totalement inenvisageable.

[2] Les exemples sont extraits de `http://www.lirmm.fr/~schwab`

Notre première hypothèse forte est qu'il est possible d'automatiser cette tâche grâce à un apprentissage basé sur des informations extraites de diverses sources (dictionnaires, listes de synonymes, indexations manuelles, recherches web, ...). L'essentiel de l'apprentissage principal se fait sur des dictionnaires à usage humain. Dans la perspective où nous plaçons cet article, dans un espace dimensionné en fonction d'une hiérarchie de concepts[3], un amorçage du système d'apprentissage est nécessaire. Il s'agit d'affecter des vecteurs à un nombre réduit d'entités qui sont choisies en fonction de leur fréquence en langue et/ou de leur polysémie. La taille du noyau est très réduite (un millier de termes environ) et les éléments de ce noyau considérés comme pertinents. À partir de ce noyau, le processus d'apprentissage peut débuter. La méthode d'analyse construit, à partir de vecteurs conceptuels déjà existants et de nouvelles définitions, de nouveaux vecteurs. L'idée est qu'à partir d'un noyau réduit d'items pertinents, un apprentissage sur des définitions permet de créer une cohérence entre les vecteurs et donc de générer une base d'items pertinents.

2.2.2 Analyse multi-sources

L'analyse des définitions pose un certain nombre de problèmes quant à la lecture du sens. C'est le cas du métalangage, c'est à dire le langage utilisé pour structurer le dictionnaire. Ce dernier est aisément utilisable lorsqu'il s'agit de récupérer les catégories grammaticales des items mais certaines constructions de définitions sont difficilement compréhensibles sans compétence métalinguistique. Comment savoir que *en parlant* est du métalangage dans une définition de l'item ‹*aboyer*› comme *Crier, en parlant du chien.*? Pour pallier de tels manques définitoires, nous utilisons diverses sources lexicales. Il s'agit de tempérer statistiquement les diverses incohérences locales. Ainsi, si une définition est mal formée (donc difficilement analysable correctement) une autre définition, mieux formée et provenant d'une autre source pourra corriger l'effet de la première. L'utilisation de multi-sources permet aussi de maximiser la probabilité de récupérer une définition pour certains termes qui sont dans certains dictionnaires et pas dans d'autres. On ne trouve pas, par exemple, ‹*liturgiste*› dans (Larousse, 2001) mais on le trouve dans (Robert, 2000).

2.2.3 Apprentissage permanent

Pour analyser un certain nombre de documents, en particulier les journaux, il est souvent nécessaire de savoir à quoi correspondent des néologismes, qui sont certaines personnalités ou encore quel est le domaine d'activité d'une entreprise. Par exemple, dans un texte, l'usage du nom de l'entreprise ‹*Usinor*› indique vraisemblablement un contexte axé sur le traitement de l'acier. Les diverses sources et en particulier le web par les serveurs d'informations (*Le Monde, Libération,* ...) présentent ces nouveautés.

Il est, de plus, difficile de penser que la base de vecteurs deviendrait cohérente dès la première passe. Il est vraissemblable que des mots clés d'une définition n'aient pas encore été appris par le système lors de son analyse. La convergence des vecteurs vers une position quasi-stable ne pourra se faire que dans un nombre de cycles qu'il est impossible de déterminer à l'avance mais qui est fonction de l'ordre d'apprentissage des items et de leur définition. Ces deux raisons, la variabilité lexicale et l'impossibilité de véritablement stabiliser une base, nous ont conduits à considérer cette troisième hypothèse forte : la base est en apprentissage permanent.

[3]Notre équipe travaille parallèlement sur la non-utilisation d'une telle hiérarchie, sur ses conséquences et ses différences avec l'approche "classique" présentée ici.

3 Vers une société d'agents

Notre objectif est la création d'un véritable système permettant l'apprentissage des vecteurs et leur exploitation. Il s'agit de récupérer des définitions, les analyser à l'aide des vecteurs déjà calculés afin d'en fabriquer de nouveaux. L'analyse simple des définitions peut ne pas toujours suffire à améliorer la cohérence des vecteurs (difficulté d'analyser certaines tournures, problèmes de métalangage, ...). Plusieurs solutions sont alors possibles comme par exemple, l'utilisation des relations sémantiques existant entre les items (synonymie (Lafourcade et Prince, 2001), antonymie (Schwab et al., 2002), hypéronymie, ...). D'autres solutions peuvent etre envisagées.

L'exploitation des vecteurs peut aussi être plurielle : utilisation pour la désambiguïsation sémantique, annotation, transfert lexical, recherche d'informations. À la fois pour l'exploitation et l'apprentissage des vecteurs conceptuels, il est donc nécessaire de pouvoir facilement ajouter des modules apportant tel ou tel service. C'est une des raisons pour lesquelle notre vision de l'architecture nécessaire s'est rapidement rapprochée des systèmes multi-agents (SMA).

3.1 SMA et TALN

Les SMA sont issus de l'intelligence artificielle distribuée (IAD) qui répartit l'intelligence dans des agents. Tout ou partie de cette intelligence est la conséquence de leur intéraction (phénomène d'émergence). Un agent est une entité physique ou virtuelle (virtuelle dans notre cas) capable d'agir sur son environnement (les autres agents), qui peut communiquer directement avec d'autres agents, qui possède des ressources propres, qui est capable de percevoir son environnement, qui possède des compétences et offre des services. On peut distinguer deux types d'agents (Ferber, 1995):

- *les agents réactifs* : les systèmes à agents réactifs suivent l'hypothèse que le système peut avoir un comportement global intelligent sans que les agents soient nécessairement intelligents individuellement. La communication des agents réactifs se fait par diffusion d'un signal dans l'environnement. Un exemple type d'agent réactif est celui des fourmis dont les actions se coordonnent afin de résoudre des problèmes complexes tel que ceux de la recherche de nourriture, construction de nid, ...

- *les agents cognitifs* : chaque agent possède sa propre base de connaissance, c'est à dire l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche. Les communications se font ici par envoi de messages entre agents et éventuellement tractations, négociations entre eux.

Nous nous situons clairement ici dans la deuxième approche, au moins, à l'échelle globale du système (cf 3.2.1). Celle-ci est utilisée déjà depuis longtemps dans le domaine des langues naturelles. Dès le début des années 70, des travaux en IAD ont été effectués sur la compréhension automatique de la parole (HEARSAY-II (Erman et al., 1980)). Plus récemment, (Lebarbé, 2001) utilise cette approche pour l'analyse syntaxique et (Menézo et al., 1996) pour la détection d'erreurs. Parmis les autres travaux, certains se rapprochent de ce que nous voulons faire, une architecture modulaire permettant l'apprentissage de données et leur utilisation. On peut citer le système CARAMEL (Sabah, 1990) ou le système TALISMAN (Stefanini et al. 1992).

Les caractéristiques de notre système sont induites par les caractéristiques générales des agents et les moyens de communiquer entre eux, c'est à dire leur interaction.

3.2 Agents

Nos agents ont des caractéristiques à la fois conceptuelles (les hypothèses que nous suivont pour les créer) et techniques (comment ils sont implémentés) .

3.2.1 Caractéristiques conceptuelles : hypothèses

Ce sont ces caractéristiques qui développent le caractère cognitif de nos agents, c'est à dire la manière dont ils "raisonnent".

- *Vision récursive des agents* : Notre système peut être vu à différentes échelles. Même si nous nous situons clairement dans une approche cognitive au niveau global de notre système, chaque agent peut être lui-même composé d'agents réactifs dont l'effet émergeant fera l'objet de transmission aux autres agents par envoi de messages. Par exemple, pour l'analyse d'un texte, un agent peut utiliser un système de fourmis (agents réactifs) sur l'arbre morpho-syntaxique correspondant afin de désambiguïser les feuilles et faire émerger le vecteur global du texte (Lafourcade, 2003) qui sera envoyé à l'agent demandeur.

- *Apprentissage par renforcement, double boucle* :

  Chacun des agents possède sa propre base de connaissance qu'il modifie au gré de ses expériences et de ses intéractions avec les autres agents. À chaque requête, les agents tirent parti des informations reçues pour modifier leurs connaissances avant de répondre à la requête. Par exemple, l'agent d'apprentissage peut extraire d'une définition une liste d'antonymes. Il va demander aux agent spécialistes de l'antonymie de lui fournir les vecteurs correspondants. Ces agents vont donc permettre une amélioration générale de la cohérence de la base. Parallèlement, ces agents vont utiliser les informations lexicales reçues du système pour modifier leurs méthodes de calcul (Schwab et al., 2002). Ainsi, les agents peuvent fournir de meilleurs résultats. Le système global s'enrichit de l'apport des agents qui eux-même s'enrichissent du système (cf. fig. 1). Ce principe est semblable à l'apprentissage dans le système nerveux central (SNC). Les neurones constituent un réseau qui se renforce à mesure qu'il est utilisé et il sera d'autant plus utilisé qu'il sera renforcé. Ce principe est connu sous le nom de *double boucle* (Lecerf, 1997).

- *Unicité des agents* : Au cours de sa vie, chaque agent est unique. Plusieurs agents peuvent pourtant avoir un rôle identique, être spécialiste d'un même domaine. Par exemple, plusieurs agents peuvent être chargés de l'analyse du français ou être spécialistes de la même relation sémantique. Ces agents peuvent être conceptuellements différents, c'est à dire qu'ils utilisent chacun une méthode particulière de résolution du problème (l'analyse sémantique d'un texte peut se faire, par exemple, par propagation d'un vecteur conceptuel dans l'arbre morpho-syntaxique ou par émergeance grâce à des agents réactifs de type fourmis) ou être simplement des clones (exactement le même code source). Dans ce dernier cas, l'unicité des agents est la conséquence de l'expérience différente acquise par chaque agent en fonction des données rencontrées ou des requêtes reçues. Il est par exemple difficile de savoir si un ordre particulier dans l'apprentissage des vecteurs leur

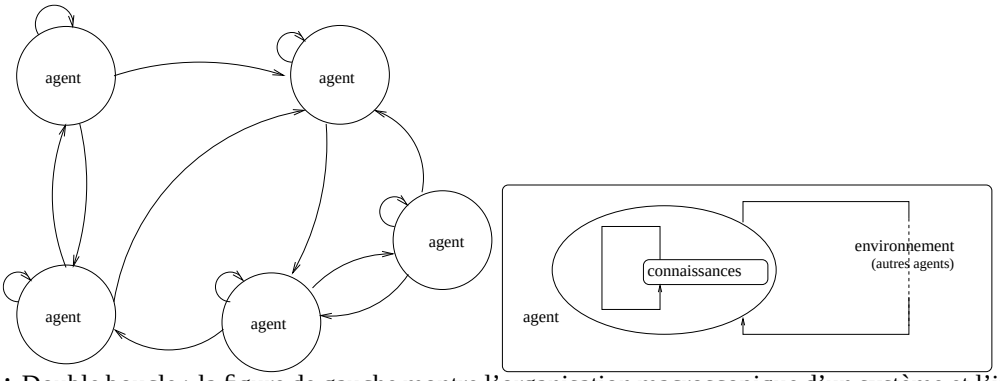

Figure 1: Double boucle : la figure de gauche montre l'organisation macroscopique d'un système et l'interaction des différents agents entre eux. La figure de droite présente l'organisation microscopique d'un système, la "vision du monde" d'un agent. Les données extérieures sont utilisées pour améliorer la propre base de connaissance de l'agent et l'agent améliore les données extérieures (double boucle)

assurerait une convergence plus rapide. Un agent d'apprentissage révise donc périodiquement de manière aléatoire chaque item de sa base. Un autre agent de ce type acquiert une expérience distincte puisqu'il analyse les items dans un ordre différent. Il en est de même des agents spécialistes des relations sémantiques puisqu'ils peuvent ne pas rencontrer les mêmes couples de termes ou les rencontrer plus ou moins souvent.

- *expertises en concurrence* : lorsqu'un agent est confronté à un problème, il peut demander à d'autres agents de l'aider. Ces agents lui donnent un certain nombre de solutions qu'il lui revient de choisir voire si possible de combiner. Par exemple, si un agent d'analyse sémantique rencontre un schéma particulier, il pourra demander aux agents experts en antonymie, hypéronymie ou méronymie si ce schéma peut les caractériser. De même, il peut demander un service à des agents spécialistes d'un même domaine. Dans les deux cas, l'agent demandeur considèrera toutes les réponses obtenues et leur attribuera plus ou moins de crédit en fonction de la compétence de l'agent.

3.2.2 Caractéristiques techniques

Ce sont les caractéristiques matérielles des agents, la manière dont ils sont implémentés et comment ils communiquent.

- *Possibilité de distribuer sur plusieurs machines* : Les systèmes TALN ont toujours été consommateurs de ressources systèmes importantes dues aux données à stocker (la taille du lexique est d'au moins 100000 entrées pour une langue comme le Français) et aux calculs souvents lourds. Chaque agent peut se trouver sur une machine, ainsi, il pourra pleinement utiliser les ressources matérielles disponibles. La communication entre agents se fait donc par accès réseaux sur le modèle client-serveur.

- *modularité* : La modularité est une caractéristique principale des architectures agent. Un service global résulte de la coopération de chaque agent qui effectue une sous-tâche moins complexe. Les avantages génie-logiciel sont nombreux : un développeur peut facilement créer un agent dans le langage informatique de son choix, pourra aisément le tester et l'améliorer indépendament des autres. L'ajout d'un programme déjà existant est simplifié. Dans notre application, par exemple, il a été extrèmement facile de créer un agent d'analyse morpho-syntaxique qui n'est qu'une interface à l'analyseur SYGMART.

La modularité est une des caractéristiques fondamentales pour notre système puisque en plus de ces avantages, il est facile de réunir plusieurs systèmes (un gérant le français, un autre l'anglais, par exemple) pour n'en obtenir qu'un.

3.3 Gestion et communication des agents

La gestion des agents se fait par un superviseur. Lors de sa création, chaque agent adresse au superviseur son identifiant, son rôle, éventuellement sa langue[4], ainsi que la machine et le port sur lequel il écoute. Le superviseur accepte la création de l'agent si aucun autre agent encore actif ne présente cet identifiant. Plusieurs types de communications sont possibles : (1) communication directe entre agents : comme pour le *point à point* (*peer to peer*), un agent demande au superviseur l'adresse particulière d'un agent puis communique directement avec lui. C'est le cas, par exemple, d'un agent *contextualiseur* qui a besoin de récupérer très souvent les informations sur les lexies (cf 4.1) que stocke la *base de vecteurs*. (2) Communication par l'intermédiaire du superviseur : Suivant la requête, les messages peuvent être envoyés à un agent, à tous les agents ayant un certain rôle, à tous les agents d'une langue ayant un certain rôle et même à tous les agents. En pratique, le message est envoyé au superviseur qui se charge de l'envoyer à ses destinataires. Chaque agent qui a reçu le message y répond en apportant une réponse ou en signifiant sa non-compétence dans ce domaine.

4 Exemple de société d'agents

4.1 Agents implémentés

À l'heure actuelle, notre système compte un certain nombre d'agents qui peuvent avoir les rôles suivants :

- *Base* : ce sont des agents dont le but est de conserver et de restituer les données nécessaires à l'apprentissage (définitions, vecteurs conceptuels).
- *Contextualiseur* : cette sorte d'agents est capable de calculer le vecteur conceptuel d'un item en fonction de contextes sémantique et morphologiques. En pratique, il fait une somme pondérée des vecteurs calculés pour chaque définition du terme en fonction d'un vecteur contexte, d'informations morphologiques et statistiques (fréquence en corpus).
- *Analyseur morpho-syntaxique* : Il s'agit de l'analyseur SYGMART(Chauché, 84). Pour un texte, il nous renvoie l'arbre morpho-syntaxique correspondant.
- *Analyseur sémantique* : avec l'aide de l'agent d'analyse morpho-syntaxique et l'aide de l'agent contextualiseur, l'agent d'analyse sémantique calcule le vecteur correspondant à un texte. Ce texte, dans le cas d'un apprentissage est une définition de dictionnaire dont l'agent d'apprentissage souhaite le vecteur (cf. fig. 2).
- *Apprentissage* : Cet agent gère l'apprentissage des vecteurs conceptuels. Il est directement aidé dans cette tâche par des agents d'analyse sémantique ainsi que par les agents extracteurs de définitions.

[4] certains agents peuvent être indépendants de la langue, c'est le cas, par exemple, d'agents dont les données ne sont que des vecteurs.

- *Agents extracteurs de définitions* : ces agents ont pour rôle de récupérer les définitions correspondant à des items et de les fournir à l'agent d'apprentissage.
- *Agents relations sémantiques* : ces agents sont des experts des relations sémantiques comme la synonymie, l'antonymie, l'hypéronymie ou toute autres fonctions lexicales.

Ces agents sont accessibles en ligne à l'adresse `http://www.lirmm.fr/~schwab`

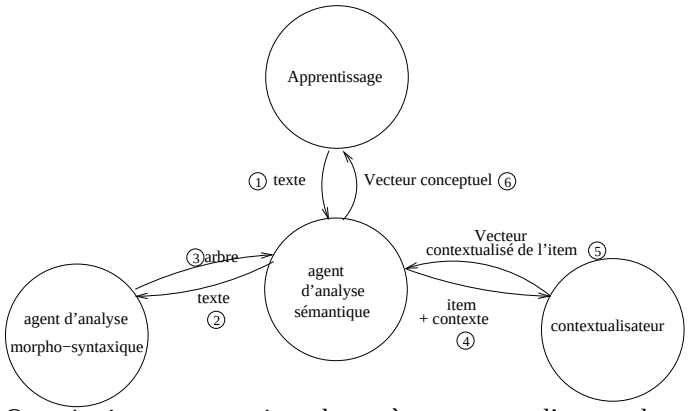

Figure 2: Organisation macroscopique du système au cours d'une analyse sémantique.

4.2 Exemple d'interaction entre agents, apprentissage du vecteur d'un item

Pour calculer le vecteur conceptuel correpondant à un item et à ses définitions, l'agent d'*apprentissage* demande à un agent *extracteur de définitions* de lui fournir les définitions de cet item qu'il a récupérées en explorant le web ou des dictionnaires à usage humain. Le texte de chaque définition est donné à l'agent *analyse sémantique* (1) qui, à partir de l'arbre morpho-syntaxique obtenu grâce à l'agent *analyseur morpho-syntaxique* (2) (3), calcule le vecteur conceptuel correspondant à la définition (6). L'agent *contextualiseur* (lui-même aidé par la *base de vecteur*) et éventuellement des agents experts en *relations sémantiques* collaborent avec lui dans cette tâche (4) (5). L'agent *apprentissage* récupère chaque vecteur des définitions et utilise l'agent *contextualiseur* pour obtenir le vecteur global de l'item. L'*apprentissage* donne alors à la *base de vecteurs* les nouveaux vecteurs calculés.

5 Conclusion et Perspectives

Dans cet article, nous avons exposé les différentes hypothèses que nous avons considérées pour construire une base lexicale de vecteurs conceptuels (génération automatique, analyse multi-sources, apprentissage permanent). Ces hypothèses ainsi que des considérations techniques telles que la modularité et la possibilité de distribuer sur plusieurs machines nous ont amené à adopter une architecture de type multi-agents. Nous avons aussi présenté les caractéristiques conceptuelles fondamentales de nos agents : une vision récursive des agents, l'unicité des agents, la manière dont les agents partagent leur expertises et le principe le plus important, celui de la double boucle. Les agents utilisent les données du système (les vecteurs) pour améliorer leur base de connaissance et agissent sur cette base pour améliorer sa cohérence. Nous avons finalement présenté certains agents qui ont été implémentés et sont accessibles en ligne. Dans la suite de nos travaux, nous allons continuer à travailler sur chaque agent pris individuellement afin d'améliorer leur tâche respectives. Nous allons aussi essayer d'améliorer leur mode de communication en analysant les moyens techniques qui permettraient aux agents de seconder un autre agent sans que celui-ci ait à demander de l'aide.

Ces travaux n'auraient pu exister sans l'aide précieuse de Mathieu Lafourcade et Violaine Prince.

Références

Chauché J., *Un outil multidimensionnel de l'analyse du discours.*, Proc of COLING'84 (2-6 July 1984 Stanford University, California), pp 11-15,1984.

Chauché J., *Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance.* TAL Information, 31/1, pp 17-24, 1990.

Deerwester S. et Dumais D., Landauer T., Furnas G., Harshman R., *Indexing by latent semantic analysis.* In Journal of the American Society of Information science, 1990, 416(6), pp 391-407.

Erman L., Hayes-Roth F., Lesser V. et Reddy D. *The HEARSAY-II Speech Understanding System : Integration Knowledge to Resolve Uncertainly.* ACM Computing Surveys, 12, 1980.

Lafourcade M. *Algorithmes "fourmis" et TALN. http://www.lirmm.fr/~lafourca/ML-research/directions/TALN-algo-fourmi/TALN-algo-fourmi.html*

Lafourcade M. et Prince V. *Synonymies et vecteurs conceptuels.* Proc. of Traitement Automatique du Langages Naturel (TALN'2001), Tours, France, Juillet 2001 pp 233-242.

Lafourcade M . *Lexical sorting and lexical transfer by conceptual vectors.* Proc. of the First International Workshop on MultiMedia Annotation (Tokyo, Janvier 2001) 6 p.

Larousse. *Le Petit Larousse Illustré 2001*. Larousse, 2000.

Larousse. *Thésaurus Larousse - des idées aux mots, des mots aux idées.* Larousse, ISBN 2-03-320-148-1, 1992.

Lebarbé T., *Vers une plate-forme multi-agents pour l'exploration et le traitement linguistique*, Proc. of Traitement Automatique du Langages Naturel (TALN'2001) (Tours, France, Juillet 2001).

Lecerf C., *Une leçon de piano ou la double boucle de l'apprentissage cognitif.*, revue Travaux et Documents, n°3-1997, Université Paris 8, Vincennes Saint-denis, Mars 1997.

Lehmann A. et Martin-Berthet F. *Introduction à la lexicologie. Sémantique et morphologie*, Paris, Dunod (Lettres Sup), 1998.

Mel'čuk I., Clas A.et Polguère A. *Introduction à la lexicologie explicative et combinatoire.* , éditions Duculot, 1995.

Morin, E. *Extraction de liens sémantiques entre termes à partir de corpus techniques.* Thèse de doctorat de l'Université de Nantes, 1999.

Menézo J., Genthial D. ET Courtin J., *Reconnaissances pluri-lexicales dans CELINE, un système multi-agents de detection et correction des erreur*, NLP+IA 96, Moncton-Canada, pp 174-180.

Ferber J. *Les systèmes multi-agents. Vers une intelligence collective.* InterEditions, 1995. ISBN 2-7296-0665-3.

Stéfanini M-H, Berrendonner A., Lallich G., Oquendo F. *TALISMAN : un système multi-agents gouverné par des lois linguistiques pour le traitement de la langue naturelle* Coling 92, Nantes, 22-29 Juillet 1992.

Le Nouveau Petit Robert, dictionnaire alphabétique et analogique de la langue française. Hachette, 2000.

CARAMEL : Un système multi-expert pour le traitement automatique des langues. Modèles linguistiques Tome XII Fascicule 1.

Salton G. et MacGill M.J. *Introduction to modern Information Retrieval* McGraw-Hill, New-York, 1983.

Schwab D, Lafourcade M et Prince V. *Amélioration de la représentation sémantique lexicale par les vecteurs conceptuels : le rôle de l'antonymie*, actes de JADT 2002, Saint-Malo, Mars 2002 .

Schwab D, Lafourcade M et Prince V. *Vers l'apprentissage automatique, pour et par, les vecteurs conceptuels de fonctions lexicales. L'exemple de l'antonymie*, actes de TALN 2002, Nancy, Juin 2002.