
EXPRESSION LANGAGIÈRE AMBIGUË ET MODELISATION COGNITIVE SYMBOLIQUE

UN MODÈLE INFORMATIQUE DE TRAITEMENT DE LA POLYSÉMIE D'USAGE

Sylvain Surcin

L.R.I.A., Université Paris 8
21, rue Baudelique, F-75018 Paris, France
Mél : surcin@lcr.thomson-csf.com

1. Objectifs

Cette thèse se situe dans le domaine de la sémantique computationnelle. Elle a pour but l'étude exploratoire de certains phénomènes d'ambiguïté lexicale que je regroupe sous la dénomination de polysémie d'usage (par exemple, dans "Le marché est dynamique aujourd'hui", *marché* désigne à la fois l'institution, la bourse, ses tendances, ses acteurs...), et la mise au point d'un prototype de traitement automatique de ces ambiguïtés par une machine.

L'objectif est de concevoir un système informatique permettant à une chaîne logicielle d'analyse automatique de textes en contexte ouvert de ne pas se trouver en situation d'échec lors de l'analyse d'expressions ou de mots ambigus relevant de la polysémie d'usage. Ces ambiguïtés se révèlent non résolubles par nature, car elles correspondent à une multiplicité d'interprétations que l'on ne peut discriminer les unes des autres, étant donné qu'elles s'avèrent toutes compatibles avec le contexte d'analyse. A moins de leur réserver un traitement particulier, un système automatique doit donc, pour éviter une situation d'échec, opérer un *choix arbitraire* entre les différentes interprétations possibles afin de n'en sélectionner qu'une. Or cette stratégie conduit, dans le cas de la polysémie d'usage, à une perte d'information qui a de grandes chances de provoquer un échec ultérieur de l'analyse.

La polysémie d'usage est une situation dans laquelle les différentes interprétations possibles de l'expression ambiguë diffèrent entre elles par des traits caractéristiques locaux n'appartenant pas à la définition stricte (intensionnelle) de l'expression en question. Ces traits sont le produit des *usages* de l'expression dans la langue en un temps et en un lieu définis, *i.e.* des références à des données socioculturelles locales qui sont passées implicitement dans la langue. De plus, une caractéristique essentielle de la polysémie d'usage est la *simultanéité* des interprétations valides : plusieurs interprétations sont à conserver en même temps, sous peine de perdre des informations potentiellement pertinentes.

En plus de l'objectif de robustesse, nous nous fixons un objectif de *pertinence relative* : le modèle doit conserver un maximum d'interprétations pertinentes simultanément. Cela afin de laisser la possibilité à un analyseur sémantico-pragmatique en aval la possibilité de conserver tous les aspects évoqués par l'expression ambiguë (au prix d'une analyse non-déterministe si cet analyseur fonctionne avec des structures sémantiques non ensemblistes).

2. Démarche

Je pars d'une réflexion sur la nature des ambiguïtés lexicales, leurs descriptions par la linguistique, et leurs traitements par le traitement automatique des langues naturelles. Connaissant les difficultés de traitement provoquées par les ambiguïtés lexicales, la première étape est de questionner la linguistique, et pour chaque réponse obtenue, d'examiner les techniques employées en informatique pour résoudre ou contourner les problèmes.

Les ambiguïtés sont expliquées en linguistique par les théories de l'*homonymie* (ou correspondance formelle fortuite entre des sens indépendants) dans la linguistique transformationnelle-générative (cf. Chomsky, Katz et Fodor), de l'*indétermination* (ou effacement de la référence devant les signes, cf. Martinet, Weydt, François) issue du structuralisme radical, et enfin de la *polysémie* (ou multiplicité des "effets de sens" pour un signe unique) chez certains structuralistes européens (cf. Le Goffic, Fuchs, Tesnière, Pottier, Culioli). Un certain nombre d'ambiguïtés sont efficacement traitées en informatique grâce à l'application des deux premières théories. Mais beaucoup d'autres sont encore insolubles, et c'est pourquoi je me tourne vers la théorie de la polysémie pour tenter de les traiter.

La polysémie est elle-même un vaste domaine, et après m'être muni d'outils linguistiques théoriques pour la caractériser et l'étudier, je propose un classement des différentes formes de polysémie en fonction à la fois des traitements qui leur sont appliqués en informatique, et de leurs caractéristiques linguistiques. Cela m'amène à distinguer :

- la *polysémie fonctionnelle* (proche de l'homonymie et bien traitée en informatique),
- la *polysémie d'acception* (qui commence déjà à faire échouer les techniques classiques de restriction de la sélection des signifiés compatibles avec le contexte)
- la *polysémie d'usage*, qui contredit toute volonté d'appliquer une quelconque stratégie de *résolution* des ambiguïtés. Celle-ci n'est qu'effleurée en linguistique. Or, elle s'avère très présente dans le langage quotidien, mais aussi dans les langages de spécialité.

La première phase de cette étude consiste à caractériser linguistiquement la polysémie : quels sont ses sous-types, ses occurrences, leurs points communs et leurs différences ? La complexité du phénomène me pousse à m'interroger sur le choix du paradigme sémantique dans lequel la décrire. Une brève étude des ressources descriptives offertes me conduit à préférer la sémantique différentielle aux cadres, plus traditionnels en intelligence artificielle, de la sémantique référentielle et de la sémantique inférentielle. C'est dans ce cadre que j'aborde les mécanismes qui sont au cœur de la polysémie d'usage : la *connotation*, la *profondeur sémantique* et la *continuité sémantique*.

Puis je cherche, parmi les systèmes existants en traitement automatique des langues naturelles, des principes de base pour constituer un modèle de traitement de la polysémie d'usage. Au terme de cette revue de l'existant, je choisis de constituer ce modèle en tant que *lexique dynamique*. Je m'inspire plus particulièrement du modèle EDGAR de Prince, conçu dans le cadre du traitement de la *polysémie des mots courants*.

Pour finir cette phase, j'ai mené une expérience destinée à fournir des renseignements complémentaires à la sémantique différentielle, qui apporte peu de données opératoires. Cette expérience consiste en l'observation d'une équipe de traducteurs professionnels face à des ambiguïtés lexicales de type polysémie d'usage. Je pars pour cela de l'hypothèse que la traduction est une forme observable d'interprétation. J'en tire un modèle fonctionnel qui a guidé la conception de mon modèle formel, ainsi que certains aspects de l'algorithme d'interprétation.

3. Le modèle PELEAS

Le modèle que j'ai conçu, PELEAS (pour *Pyramids and Ellipsis as Lexical Entries in Ambiguous Sentences*), a pour vocation de traiter les trois formes de polysémie recensées : polysémie fonctionnelle, polysémie d'acceptation et polysémie d'usage.

C'est un modèle symbolique hybride : chaque entrée du lexique correspond à une structure de réseau hiérarchique d'étiquettes lexicales (cinq niveaux de profondeur sémantique donnant une approximation satisfaisante du contexte). Les liens entre les étiquettes de même niveau expriment des *contraintes sémantiques* propres à l'entrée, tandis que les liens inter-niveaux expriment des *relâchements de contraintes* génériques. Ces structures sont invariantes par symétrie d'échelle en profondeur.

L'algorithme d'interprétation d'une entrée lexicale ambiguë dans un contexte donné est amorcé par des *règles de sensibilité au contexte*. Il fonctionne ensuite par *propagation d'activité symbolique* le long des liens entre les étiquettes. Il fournit en sortie un résultat exhaustif (l'ensemble de toutes les étiquettes lexicales avec leur taux d'activité finale : saillant, contredit, valide ou inhibé) et un résultat synthétique (la liste des interprétations significatives accompagnées d'une estimation de leur participation à la construction de la signification de l'entrée lexicale). Cet algorithme est de complexité linéaire au pire en fonction de la taille de l'entrée lexicale.

Le modèle PELEAS est mis en œuvre par un jeu de trois logiciels : le moteur d'inférences LightPeleas, l'éditeur d'entrées lexicales Melisande et un outil annexe, Bard. Cet outil est destiné à l'exploration de corpus pour l'extraction semi-automatique de données destinées à alimenter le lexique existant. Ces logiciels ont été développés sur un micro-système (PC sous Windows NT) en utilisant la technologie de distribution d'objets ActiveX. Leur conception utilise des techniques avancées de conception d'architectures logicielles et de spécification formelle.

4. Résultats

Le modèle PELEAS a été évalué sur un corpus composé de 110 phrases pour un lexique de 21 entrées fortement sujettes à des phénomènes de polysémie d'usage ou d'acceptation. Ces phrases ont été choisies pour les difficultés d'interprétation qu'elles comportent (slogans publicitaires, extraits de la presse écrite ou orale, extraits littéraires). Je cherche en effet à savoir ce qu'apporte PELEAS pour des situations limites, et non pas à le comparer à des outils de résolution d'ambiguïtés courantes sur gros corpus. Sur ce corpus de test, PELEAS s'est montré tout à fait robuste (aucune situation d'échec) et raisonnablement pertinent (peu de bruit et jamais de contresens). Sur des cas de polysémie fonctionnelle, il donne des résultats équivalents à ceux d'un système classique de résolution d'ambiguïté par restriction de la sélection. De plus, un test de non-régression par rapport au modèle EDGAR lui a été appliqué avec succès.

En conclusion, le modèle PELEAS s'avère plus particulièrement intéressant pour l'interprétation de jeux de mots, cumuls de sens délibérés et phrases "à tiroirs", sans être pénalisant pour des formes d'ambiguïté lexicale plus simples.

Une extension prévue est la prise en compte des *afférences locales* (interactions locales entre deux entrées ambiguës). De plus, une application envisagée est son utilisation pour l'assistance à la traduction, comme outil d'aide à la prise de décision pour des expressions techniques ambiguës.