وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة فرحات عباس – سطيف –

Université Ferhat Abbas - Sétif -

Mémoire

Présenté à la Faculté des Sciences de l'Ingénieur

Département d'Informatique

pour l'obtention du diplôme de

Magistère

Option Sciences et Technologies de l'Information et de la Communication

Par Brahim SAHRAOUI

THEME

La résolution des anaphores pronominales dans le TAL.

Soutenu le	Devant le jury d'examen :	
Dr. A Boukerram	MCA UFA de Sétif	Président
Dr. A. Refoufi	MCA UFA de Sétif	Rapporteur
Dr. M. Aliouat	MCA UFA de Sétif	Examinateur
Dr. Y. Salem	MCB UFA de Sétif	Invité

2010/2011

REMERCIEMENTS

Tout d'abord, je tiens à déclarer mes remerciements avec une profonde reconnaissance et gratitude à mon encadreur Dr Allaoua Refoufi qui, en tant que directeur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'il a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

Je remercie également le Président et les membres du jury de m'avoir fait l'honneur d'accepter de juger ce mémoire. Et tous les enseignants d'Informatique de l'université de Sétif.

Enfin, j'adresse mes plus sincères remerciements à mes parents, qui m'ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire. Je n'oublie pas tous mes proches et amis pour leur contribution, leur soutien et leur patience.

SOMMAIRE

1. Introduction

La résolution des anaphores est une des branches de recherche les plus actives du domaine du Traitement automatique des langues (TAL). Le phénomène de l'anaphore est très répandu dans les langues naturelles, c'est pour cela qu'un module de résolution est nécessaire dans presque toutes les applications du TAL. Des travaux très variés ont été effectués dans ce domaine selon des approches théoriques différentes.

La résolution consiste à trouver la référence d'un groupe nominal qui doit être interprété par rapport à un élément apparaissant avant lui dans le discours. Ce groupe nominal peut être un nom avec un déterminant (groupe nominal plein), un pronom personnel, un pronom démonstratif ou réfléchi. Le problème de la résolution comprend deux parties. La première est la résolution de la coréférence, qui consiste à établir une relation entre deux groupes nominaux pleins qui font référence à un même élément dans le discours. La deuxième consiste en la recherche de la référence d'un pronom, élément qui doit obligatoirement être interprété selon son contexte. Cette deuxième opération est ce qu'on appelle la résolution d'anaphore pronominale.

Les pronoms anaphoriques peuvent reprendre des éléments linguistiques de nature catégorielle diverse: des noms, des groupes nominaux (anaphore individuelle), mais aussi des verbes, ou des phrases (anaphore abstraite). Le but de l'étude qui suit est d'examiner le phénomène d'anaphore et de formuler un algorithme pour la résolution.

La logique de base de notre algorithme est basée en grande partie sur l'algorithme de Lappin et Leass. La procédure de résolution consiste à examiner le texte pour l'extraction des pronoms et les groupe nominaux qui les précèdent. Ensuite, pour chaque pronom on éliminera les référents du discours pour lesquelles une expression anaphorique ne peut se référer, puis en sélectionnant l'antécédent optimal parmi les candidats qui restent.

Compte tenu des difficultés dans le domaine, il est sans doute temps de remettre en question la définition de la tâche. Notre préoccupation primaire est de cerner les éléments importants en rapport avec les reprises anaphoriques et concevoir une tâche de TAL qui ne se prête pas à des accommodations au moment de l'évaluation des systèmes.

Pour ce faire. Dans le premier chapitre on entamera la définition des divers termes du domaine: la référence, la coréférence, l'anaphore et le facteur prise en compte dans la résolution des anaphores. Nous présenterons également quelques exemples pour clarifier les concepts. Dans le deuxième chapitre on expliquera les algorithmes les plus connus dans le monde du TAL qui se basent sur des concepts syntaxiques, morphologiques et sémantiques. On présentera dans le troisième chapitre une procédure de résolution automatique des liens anaphoriques basée sur une description syntaxique, et l'appliquera sur un certain jeu de test. Enfin on compare le résultat de notre algorithme avec les algorithmes vu en état de l'art et nous parlerons des avantages et des inconvénients de cette approche.

2. La résolution des anaphores

Les définitions dans le domaine des anaphores ne sont pas aussi figées que l'on pourrait le croire. Il est en effet difficile de bien cerner la nature des éléments qui participent à une relation anaphorique, ainsi que la nature de la relation elle même. Le flou terminologique qui en résulte a des conséquences sur la définition des tâches de traitement automatique de ces phénomènes.

2.1. LA REFERENCE

Décrire la façon dont les mots de la langue font référence au monde qui nous entoure, que ce soit le monde que nous percevons au moyen de nos sens ou celui des idées et concepts abstraits, constitue une grande question de philosophie. Comment définir la relation qui entre en jeu lorsque nous parlons d'un objet concret ou d'une idée abstraite, et comment décrire la façon dont elle relie nos mots à la réalité ? Le but est ici de définir la notion de référence.

Les mots, ou plus précisément les unités lexicales, des langues naturelles sont classifiés selon leur catégorie syntaxique. Les unités lexicales se combinent (en fonction de leurs catégories) pour former des groupes syntaxiques (ou syntagmes). Les groupes de types différents ne font pas référence au monde de la même façon. Par exemple, les groupes verbaux ne représentent pas le même type d'entités que les groupes nominaux. Ces deux sortes de référence sont distinctes.

Le référent d'une unité lexicale est la partie du monde ou l'entité à laquelle cette unité est associée. Celui-ci peut être de nature abstraite ou concrète. Les unités lexicales de la langue ne peuvent avoir pour référent n'importe quelle entité du monde : les groupes nominaux un homme et une femme, par exemple, désignent des entités différentes. La langue impose des contraintes sur la référence de ses unités lexicales et les distingue en spécifiant le type d'entité que chacune peut désigner. Un type correspond à l'ensemble des propriétés qu'une entité doit représenter pour être le référent d'une unité lexicale donnée. Par exemple, pour être le référent de l'unité lexicale *homme*, une entité doit posséder un ensemble de propriétés, parmi

lesquelles se trouvent *être humain* et *de sexe masculin*. Le type d'une unité lexicale reflète donc la réalité, mais ne réfère pas directement au monde.

Milner distingue deux types de référence : la référence virtuelle qui correspond à l'ensemble des propriétés qu'une unité lexicale possède tandis que la référence actuelle correspond à l'entité du monde associée à cette unité.[01, Bittar]

Avec cette définition de la notion de référence à l'esprit on procédera à une considération de deux relations qui en dépendent : la coréférence et l'anaphore.

2.2. LA CO-REFERENCE

En langue, il est possible que deux unités lexicales ou séquences d'unités fassent référence à la même chose. La co-référence est une relation linguistique qui s'établit entre deux unités lexicales qui ont la même référence. De la même manière qu'on distingue deux types de référence, on distingue co-référence virtuelle et co-référence actuelle.

2.2.1 La co-référence actuelle

Quand deux unités lexicales partagent un même référent, il y a co-référence actuelle. Dans ce cas elles désignent une même entité dans le monde. La co-référence actuelle est une relation symétrique entre deux unités et implique l'identité matérielle absolue des référents, mais pas forcément l'identité des unités elles-mêmes :

[Ferhat_Abbas]$_i$ est le Premier président du Gouvernement provisoire de l'Algérie. [Ce_politicien]$_i$ été un pharmacien à Sétif.

Dans cet exemple, il y a co-référence actuelle entre les groupes nominaux *Ferhat_Abbas* et *ce politicien* car ils réfèrent à la même entité, même si les unités lexicales ne sont pas identiques.

2.2.2 La co-référence virtuelle

Lorsque deux unités lexicales distinctes ont les mêmes propriétés, elles sont en relation de co-référence virtuelle. Cependant deux unités lexicales ne

peuvent avoir exactement le même ensemble de propriétés à moins d'être identiques.

Comme le dit Milner, on peut douter de l'existence de la synonymie lexicale absolue dans les langues naturelles. Cela voudrait dire que la relation de co-référence virtuelle ne peut exister qu'entre une unité ayant des propriétés lexicales spécifiées et une autre qui n'en a pas. Ce sont les pronoms de troisième personne qui sont dépourvus de référence virtuelle. Il n'est pas possible d'attribuer au pronom *en*, hors contexte, un ensemble de propriétés qui permettrait de lui associer un type d'entité dans le monde. En revanche, en contexte il devient interprétable :

[Prolog]$_i$ est un langage de programmation logique. [Il]$_i$ permet de créer des applications intelligentes.

Dans cet exemple on a un groupe nominal, *Prolog*, pourvu de références virtuelle et actuelle. Le pronom *il* entretient une relation de co-référence virtuelle avec le groupe nominal. Le pronom possède donc toutes les propriétés lexicales du groupe nominal *Prolog* (co-référence virtuelle). Il désigne également les mêmes entités du monde (co-référence actuelle).

On a introduit deux types de relation de co-référence qui dépendent de la notion de référence. On procède maintenant à une discussion de la deuxième relation qui nous intéresse, l'anaphore.

2.3. L'ANAPHORE

2.3.1 Définition

La définition classique donnée par Halliday et Hasan [a, Halliday et autre] est basé sur la notion de la cohésion : « l'anaphore est une cohésion (présupposition) qui se dirige de nouveau à un certain élément précédent ».
Dans un discours, à l'oral ou à l'écrit, on fait souvent référence à un même objet, fait, action ou événement de façon répétitive. Mais on ne l'évoque pas toujours de la même façon. Si on parle de *la vile de Setif*, on utilisera un pronom à chaque fois qu'on voudra en dire quelque chose : *Elle est la ville la*

plus propre dans L'Algérie. Ceci évite la répétition inutile d'informations et assure la cohérence de notre discours. L'usage de pronoms pour la reprise d'éléments mentionnés précédemment dans un discours met en jeu la relation d'anaphore.

L'anaphore existe entre deux unités lexicales quand l'interprétation de l'une nécessite la présence de l'autre. Milner considère que les pronoms sont dépourvus de référence virtuelle propre, qu'ils ne sont pas référentiellement autonomes. Cela veut dire qu'ils ne sont pas capables de déterminer leur propre référent. Ils ne désignent pas de façon indépendante des entités du monde. On a vu qu'un pronom hors-contexte n'est associé à aucun ensemble de propriétés lexicales, mais qu'il peut être interprété en contexte. Il doit obtenir sa référence virtuelle à partir d'un autre élément dans le discours.

[L'université_de_Sétif]$_i$ est une des plus grandes universités algériennes. Elle$_i$ enseigne la plupart des disciplines.

[Votre_voiture]$_j$ est très puissante. Elle$_j$ dépasse cent kilomètres à l'heure en moins de cinq secondes.

Dans le premier exemple le pronom elle possède toutes les propriétés lexicales du groupe nominal université de Sétif. Mais dans le deuxième, elle possède les propriétés lexicales du groupe nominal ta voiture.

La relation d'anaphore pronominale combine la co-référence avec une relation asymétrique de reprise qui existe entre deux termes hétérogènes (groupe nominal – pronom ou nom – pronom), l'un étant référentiellement autonome, l'autre pas. L'anaphore repose toujours sur la co-référence virtuelle; parfois elle repose également sur la co-référence actuelle :

[Hassiba_Ben_Bouali]$_i$ est une martyre de la révolution algérienne durant le colonialisme français. [Elle] était une militante et résistante durant la guerre d'Algérie,

Le pronom *Elle* est ici en relation de co-référence actuelle avec le nom propre *Hassiba_Ben_Bouali* , qui a une référence actuelle. Les deux références actuelles sont identiques. Le nom donne une référence virtuelle au pronom, qui en était dépourvu.

J'ai mangé [trois_gâteaux]$_i$ et toi tu [en]$_i$ as mangé cinq.

Dans ce cas, le pronom partage la référence virtuelle du nom *gâteaux*. En revanche, il ne partage pas la référence actuelle du groupe nominal *trois gâteaux*, car il ne s'agit pas des mêmes gâteaux. Il y a donc une relation de co-référence virtuelle entre le pronom et le nom qu'il reprend, mais pas de co-référence actuelle.

Ces deux exemples comportent une unité dépourvue de référence virtuelle, l'anaphore, et une unité lexicalement spécifiée qui la précède et qui a une référence virtuelle : l'antécédent.

Seule une unité lexicalement spécifiée peut jouer le rôle d'antécédent. Les pronoms, étant dépourvus de référence virtuelle propre, ne peuvent pas en fournir une, et ne peuvent donc pas être l'antécédent dans une relation anaphorique.

Cependant, dans les chaînes de référence, où plusieurs pronoms anaphoriques ont le même antécédent, ils entretiennent une relation de co-référence.

Il est important de ne pas oublier la distinction entre la relation d'anaphore et celle de co-référence. Contrairement à la co-référence, l'anaphore est une relation asymétrique qui existe entre un élément anaphorisé (l'antécédent) et un deuxième élément anaphorisant (l'anaphore). Les deux relations peuvent coexister, et tel est le cas quand il y a anaphore pronominale, ce qui implique toujours une co-référence virtuelle. C'est justement grâce à la co-référence virtuelle qu'on arrive à interpréter le pronom dans une relation anaphorique.

2.3.2 Les types des anaphores :

2.3.2.1 L'anaphore pronominale:

C'est une reprise d'un terme à travers différents types de pronoms tels que pronoms personnels, pronoms démonstratifs, pronoms possessifs, pronoms relatifs ou pronoms indéfinis. Les pronoms personnels « je/tu » ne sont généralement pas cité comme représentatifs de l'anaphore pronominale; par contre les pronoms personnels de troisième personne, sont des unités

présentées par la tradition comme typiquement anaphoriques. Il existe quatre groupes d'anaphore pronominale:

Totales : le substitut pronominal correspond à l'antécédent dans sa totalité (pronom personnel).

J'ai étudié [l'AUML]. [C]'est une version de l'UML orienté agent.

Partielles : le substitut pronominal ne correspond qu'à une partie de l'antécédent comme les pronoms indéfinis (quelques uns, la plupart,…).même les pronoms démonstratifs peuvent donner de tels anaphores partielles (ceux-ci, ceux-là...).

J'ai acheté [six gâteaux], j'[en] ai mangé deux sur place.

Anaphores dissociées : (divergentes), (ou virtuelles) : le pronom ne renvoie pas à l'antécédent, ni à une partie de l'antécédent, mais à un référent différent qui appartient à la même classe de l'antécédent. Référentiellement, il y a deux référents différents mais qui appartiennent à la même classe.

[Ton stylo] est beau mais [le mien] écrit mieux.

Anaphore associative (pronominale) : c'est le contexte qui permet de créer le concept

"Homme " compris dans *« on s'est mariés »*

2.3.2.2 Les anaphores nominales:

C'est une reprise d'un terme à travers un nom ou un groupe nominal. Il existe quatre groupes d'anaphore nominale:

L'anaphore fidèle: Qui est la reprise d'un nom uniquement à travers le changement du déterminant.

Louise a trouvé [un chien] dans la rue. [Le chien] aboyait sans cesse.

L'anaphore infidèle*:* est la reprise à travers des changements lexicaux. Le groupe nominal anaphorique contient des éléments autres que ceux du terme précédent.

Louise a trouvé [un chien] dans la rue. [L'animal] aboyait sans cesse.

L'anaphore conceptuelle: est la reprise d'un groupe nominal ou d'un segment qui n'apparaissent pas explicitement dans la partie précédente du

texte. Elle résume le contenu d'une phrase, d'un paragraphe ou d'un fragment de la partie du texte qui précède.

[Vous le tenez pour incapable]. Votre [préjugé] est stupide.

***L'anaphore associative**:* se base sur une relation de tout à partie.

C'était [des chevaux étranges]. [Les sabots] étaient petits, [la crinière] soyeuse.

2.3.2.3 L'anaphore verbale

Elle se réalise au moyen du verbe "faire", qui représente un verbe dénotant un processus.

On n'agit pas comme vous faites.

2.3.2.4 L'anaphore adverbiale

C'est la reprise d'un terme à travers un adverbe du type "ainsi", "pareillement" ou de l'adverbe de lieu "là".

Sa mère le priait d'aller [chez le dentiste], mais c'était justement [là] qu'il ne voulait pas aller.

2.3.2.5 L'anaphore adjectivale

C'est quand on utilise l'adjectif "tel" pour représenter une proposition précédente.

On vous a dit que la décision était sévère, je vous dis que je ne partage pas une telle opinion.

2.4. FORMES NON ANAPHORIQUES

Un point important à noter est que toutes les occurrences de pronoms n'impliquent pas forcément une relation anaphorique. Dans certains cas, les pronoms que l'on utilise dans le discours obtiennent une référence de façon déictique, c'est-à-dire par rapport au contexte d'énonciation ; parfois, ils n'ont pas de référence, ce sont des pronoms impersonnels. Les pronoms impersonnels ne sont pas référentiels. Ils occupent simplement une position argumentale (le plus souvent de sujet syntaxique) d'un verbe, sans avoir de contenu sémantique. Ils ne sont donc pas anaphoriques. Il existe également

des constructions où un pronom, sans être qualifié d'impersonnel, n'est pas anaphorique.

Dans une opération de résolution d'anaphores, il est donc nécessaire de distinguer les pronoms qui ont un antécédent potentiel de ceux qui n'en ont pas.

2.4.1 Verbes météorologiques :

Les occurrences du pronom cela/ça dont il ne faudra pas chercher l'antécédent sont celles que l'on peut considérer comme un emploi impersonnel. En français, les verbes météorologiques tombent dans cette catégorie. Les verbes tels que pleuvoir, neiger, cailler et faire beau prennent tous un sujet impersonnel, en général le pronom il.

2.4.2 Expressions figées :

On trouve certaines expressions familières qui contiennent un verbe avec un sujet impersonnel sous la forme du pronom ça. Ce sont des expressions que l'on emploie le plus souvent à l'oral comme ça barde, ça va, ça roule, ça boume etc. Ces verbes, lorsqu'ils sont employés dans des expressions de ce genre, ne peuvent en effet prendre que le pronom ça comme sujet.

2.4.3 Pronoms sujet de verbe à complétive :

Un autre cas où on peut considérer que ça est non anaphorique est quand il est sujet d'un verbe à complétive :

Ça m'étonnerait qu'il vienne.

Ça fait plaisir de te voir.

Ça lui arrive de se tromper.

2.4.4 Les relatives périphrastiques :

Ce genre de construction (aussi appelées relatives indéfinies) n'ont qu'un sens très général. Elles sont introduites par le pronom ce, invariable, qui est suivi du pronom relatif que, qui, dont ou quoi précédé d'une préposition. Le contenu sémantique du pronom n'est pas explicité :

Ce qui est rare est cher.

Ce que je dis déplaît au gens.

Ce dont je parle est très sérieux.

J'aime tout ce qui est cher.

2.4.5 Cataphore

Il est toutefois permis d'utiliser une forme pronominale vide en début de texte, afin d'attirer l'attention sur les éléments qui l'explicitent immédiatement. Ce procédé, qui lance le lecteur vers l'avant afin de lui permettre de donner un contenu référentiel à une forme vide, est exactement l'inverse de l'anaphore et s'appelle une cataphore.

Il est là, sous le sapin, le jouet dont l'enfant a tant rêvé

En bref, une cataphore est une anaphore renversée.

2.5. LES FACTEURS DE RESOLUTION

Malgré le travail considérable sur la résolution des anaphores jusqu'ici, il existe un certain nombre de questions en suspens liées à des facteurs qui forment la base des algorithmes de résolution de l'anaphore. Pour commencer, nous ne savons pas encore s'il est possible de proposer un ensemble de facteurs utilisés dans la résolution anaphore et s'il y a des facteurs que nous ne sommes pas pleinement conscients.

Les facteurs (les symptômes, les indicateurs) sont généralement divisées en des contraintes et des préférences, mais d'autres auteurs voire que tous les facteurs doivent être considérés comme préférentielles, ce qui donne une plus grande préférence à des facteurs plus restrictif et moins de préférence au moins restrictif. Mitkov [02, Mitkov] montre que la frontière entre les contraintes et les préférences est assez floue et que le traitement de certains facteurs d'une façon absolu peut être trop risqué.

L'impact des différents facteurs et / ou de leur coordination ont également été étudiés par M. Carter[10, Carter] [11, Carter]. Il fait valoir qu'une structure de contrôle souple basé sur des scores numériques attribués aux préférences permet une plus grande coopération entre les facteurs, par opposition à une architecture de profondeur d'abord plus limitée.

En plus de l'impact de chaque facteur sur le processus de résolution, les facteurs peuvent avoir un impact sur d'autres facteurs indépendants. Une question qui mérite davantage d'attention est la "dépendance mutuelle" des facteurs. Cette dépendance est défini de la façon suivante: étant donné les facteurs x et y, y est considéré comme étant à charge sur le facteur x dans la mesure où la présence de x implique y. Deux facteurs seront appelés mutuellement dépendants si chacun dépend de l'autre.

Enfin, même si un certain nombre d'approches utilisons un ensemble similaire de facteurs, les "stratégies de calcul" pour l'application de ces facteurs peut diffèrent. Le terme "stratégie de calcul" fait ici référence à la façon dont les facteurs sont utilisés, à savoir les formules pour leur application, l'interaction, poids et autres. Mitkov [14, Mitkov] a montré que ce n'est pas seulement la sélection optimale des facteurs qui importe, mais aussi le choix optimal de la stratégie de calcule.

2.6 APPLICATIONS DE LA RESOLUTION DES ANAPHORES:

2.6.1 La traduction automatique

La majorité de systèmes de traduction automatique ne traitent pas la résolution d'anaphore et leur réussie habituellement ne dépasse pas le niveau de phrase. La complexité est due aux anomalies de genre à travers les langages, pour numéroter des anomalies des mots dénotant le même concept, aux anomalies dans la transmission de genre des pronoms possessifs et anomalies dans la sélection de l'anaphore de la langue cible. Ce dernier peuvent être vus par fait que quoique dans la plupart des cas le pronom dans la langue source soit traduit par un pronom de langue cible (l'équivalent de traduction de l'antécédent du pronom de langue source au lequel correspond dans le genre et le nombre), là sont quelques langages dans lesquels le pronom est souvent traduit directement par son antécédent (Malais). En plus, des anaphores pronominales souvent sont elliptiquement omis dans le langage cible (espagnol, italien, japonais, coréen). Un autre exemple

intéressant est la traduction de l'anglais au coréenne. Les pronoms anglais peuvent être omis elliptiquement, traduit par un groupe nominal défini, par leur antécédent, ou par un ou deux pronoms coréens possibles, selon l'information syntaxique et la classe sémantique du nom auquel l'anaphore se réfère.

2.6.2 Extraction de l'information

La résolution d'anaphore dans l'extraction de l'information a pu être considérée en tant qu'élément des plus tâche générale de la résolution de coréférence qui prend la forme de fusionner des données partielles objecte les entités à peu près identiques, les rapports d'entité, et les événements décrits à différent positions de discours [08, Kameyama] .

La résolution de Coréférence a été introduite comme nouvelle tâche domaine indépendante à la 6ème Conférence de compréhension de message (MUC-6) en 1995. Ceci a provoqué l'un certain nombre traitement de projets (Srivinas et Baldwin 1996 [17, Srinivas et autres] , Gaizauskas et autres 1998, [04, Gaizauskas et autres] Kameyama 1997[08, Kameyama]) avec la résolution de co-référence dans l'extraction de l'information.

2.6.3 Résumé automatique

Le but d'un résumé automatique de texte est de produire une représentation abrégée d'un ou de plusieurs documents.

Des chercheurs dans le résumé automatique se sont intéressés à la résolution des anaphores puisque les techniques pour extraire des phrases importantes sont plus précises si des références anaphoriques des concepts/groupes nominaux sont résolues.

3. État de l’art

Le fait que la résolution d'anaphore soit un problème compliqué dans le traitement du langage naturel a attiré l'attention de beaucoup de chercheurs. Parmi les algorithmes pour la résolution, celui de J Hobbs, qui fonctionne uniquement sur base de critères syntaxiques est un des précurseurs. Et de S.Lappin et H.J. Leass qui ont formulé un algorithme qui exploite des informations morpho-syntaxiques et sémantiques. Cette approche hybride a eu un certain degré de succès et reste une des références dans le domaine.

3.1 Hobbs(1978)

J Hobbs [07, Hobbs] a élaboré un algorithme sur la base de contraintes imposées par la syntaxe de l'anglais. L'algorithme prend en entrée un arbre syntaxique complet et correct qu'il parcourt à la recherche d'antécédents en leur appliquant diverses contraintes syntaxiques et morphologiques. Au niveau intraphrastique, l'algorithme consiste en un parcours en largeur de gauche à droite avec une préférence pour l'antécédent le plus proche de l'anaphore. Un parcours en largeur est aussi effectué au niveau interphrastique, avec une préférence pour les sujets comme antécédents. En effectuant le parcours, l'algorithme fait l'inventaire des antécédents possibles, qu'il vérifie ensuite en appliquant des contraintes d'accord morphologique (traits de genre et nombre). Il applique également des contraintes syntaxiques, qui sont basées sur la condition B de la théorie du liage [04]. Ce sont notamment :

- Un pronom non réfléchi et son antécédent ne peuvent pas apparaître dans la même phrase simple.
- L'antécédent d'un pronom doit précéder ou c-commander le pronom.(voir chapitre 3.)

Les pronoms traités par cet algorithme sont les pronoms personnels he, she, it et they. Le taux de réussite global était assez élevé (le taux est le nombre d'anaphores trouver sur le nombre totale des anaphores), il est de 88.3%. Cependant, il échoue sur certains cas, tels que la reprise d'éléments phrastiques tels que :

[Salim avait des ennuis]$_i$ et il le$_i$ savait.

L'algorithme effectue aussi une recherche pour des antécédents de pronoms cataphoriques, qui apparaissent après le pronom. Cependant il ne cherche pas en dessous des niveaux des groupes nominaux ou des propositions, ce qui fait que les phrases comme la suivante ne sont pas traitées :

Noureddine$_i$ a dormi dans l'appartement de Boubakeur$_j$ avant qu'il$_j$ ne l'a viré.

L'approche de Hobbs, qui date maintenant de presque trente ans, a donné des résultats assez bons. Cependant, il est évident qu'il ne peut pas résoudre les pronoms dans certaines constructions. Les approches plus récentes, telles que celle de Lappin et Leass, prennent en compte d'autres paramètres que la syntaxe, ce qui peut aider à améliorer la couverture du système de résolution.

3.2 Lappin & Leass (1994)

S. Lappin et H.J. Leass [11, Lappin et autres] proposent un algorithme pour l'identification des antécédents nominaux de pronoms de troisième personne (he, she, they, it) et d'anaphores réflexives et réciproques (himself, herself, themselves, itself) en anglais.

L'algorithme se base sur l'utilisation d'informations de nature syntaxique et morphologique. Il utilise un modèle qui calcule dynamiquement la saillance d'un antécédent potentiel sur la base de différents facteurs. A chaque facteur est attribué un indice différent selon son utilité dans la procédure de résolution. Cette mesure de saillance pondérée est utilisée afin de classer les candidats potentiels pour déterminer une préférence. Avant l'application des mesures de saillance, les pronoms non-anaphoriques sont éliminés. Les pronoms pléonastiques, comme it dans des constructions avec un adjectif modal, sont mis à l'écart : it is necessary/essential/sufficient to/that.

Les occurrences de it avec un verbe cognitif ou d'attitude propositionnelle sont aussi exclues, car le pronom est impersonnel: it is recommended/ believed/ known/ expected that.

Ensuite, un filtre de contraintes morphologiques et syntaxiques élimine les candidats qui ne satisfont pas les contraintes de la théorie du liage ou les contraintes d'accord de genre et nombre. Ce filtre est mis en place afin d'exclure la coréférence dans les exemples suivants :

The woman$_i$ said that he$_i$ is funny.

She$_i$ likes her$_i$.

She$_i$ sat near her$_i$.

This is the man he$_i$ said Hamza$_i$ wrote about.

His$_i$ portrait of Nabil$_i$ is interesting.

Tous les référents qui forment une chaîne anaphorique sont regroupés dans des classes d'équivalence. A chaque classe d'équivalence est attribué un poids, qui correspond à la somme de tous les facteurs de saillance qui sont associés à au moins un membre de la classe d'équivalence.

Au fur et à mesure de la résolution, les valeurs de saillance des antécédents potentiels sont diminuées selon certaines règles. Dans une classe d'équivalence, c'est l'antécédent le plus saillant qui est choisi comme le référent d'un pronom.

Dans les situations où deux ou plusieurs antécédents ont la même mesure de saillance, c'est celui qui est le plus proche du pronom qui l'emporte. Les antécédents intraphrastiques ont la priorité sur ceux qui sont interphrastiques. Les facteurs de saillance utilisés dans l'algorithme sont principalement des propriétés structurales ou syntaxiques. Chaque facteur permet, selon sa pertinence, d'augmenter le score des antécédents potentiels:

– Sentence recency : exprime le fait que plus un antécédent est proche de l'anaphore, plus il est saillant. La valeur de ce facteur est diminuée de moitié pour chaque nouvelle phrase. (100)

– Subject emphasis : un antécédent en position sujet est plus saillant que ceux ayant d'autres rôles grammaticaux. (80).

– Existential emphasis : un élément nominal dans une construction existentielle (« There is... ») est saillant. (70).

– Accusative emphasis : un élément en position d'objet direct est saillant, mais pas autant qu'un sujet. (50).

– Indirect object, oblique complement emphasis : un objet indirect est moins saillant qu'un élément objet direct. (40).

– Head noun emphasis : une tête nominale dans un groupe nominal complexe est plus saillante qu'un nom non tête, qui est pénalisé.

– Non adverbial emphasis : ce facteur pénalise les GN dans des constructions adverbiales.

Une implémentation de l'algorithme a été effectuée en Prolog et testée sur un corpus de manuels informatiques. Les tests ont donné un taux de réussite de 86%, 4% de plus que l'algorithme de Hobbs sur le même corpus.

Cette approche donne d'assez bons résultats. Le système de Lappin et Leass a pour inconvénient de nécessiter diverses données d'entrée et de nombreux niveaux d'analyse. Leur algorithme fonctionne avec une analyse syntaxique et morphologique, et nécessite aussi une analyse des rôles sémantiques des groupes nominaux dans le texte. Cela rend une implémentation assez coûteuse en temps et en code. Un des algorithmes de Mitkov, que l'on examinera par la suite, a été conçu pour contourner ce problème.

3.3 Kennedy et Boguraev (1996)

La méthode de résolution proposée par Kennedy & Boguraev est la version modifié et étendue de celle développée par Lappin & Leass [10, Kennedy et autres].

Le système de Kennedy et de Boguraev n'exige pas une analyse syntaxique détaillée complètement mais il utilise la sortie d'un tagger des parties du discours, enrichie seulement avec des annotations de fonction grammaticale des éléments lexicaux dans le texte d'entrée.

La logique de base de leur algorithme est parallèle à celle de l'algorithme de Lappin et Leass.

Cependant. L'algorithme de Lappin et Leass se fonde sur l'information basée sur la configuration syntaxique, tandis que celui de Kennedy et de Boguraev, en l'absence d'une telle information, compte sur des inférences de fonction et de priorité grammaticales à déterminer la référence.

Après que les filtres morphologiques et syntaxiques aient été appliqués, l'ensemble des référents de discours qui demeurent constitue l'ensemble d'antécédents de candidat pour le pronom.

Le positionnement d'un candidat est soumis à un procédé final d'évaluation qui remplit deux fonctions :

Il diminue la saillance des candidats que le pronom précède (le cataphore est pénalisé), et augmentations la saillance des candidats qui satisfont une localité ou a état de parallélisme, qui s'appliquent aux candidats intra- phrase. Le candidat avec la plus haute saillance (le poids) est déterminé pour être l'antécédent réel ; en cas d'a la cravate, le candidat le plus proche est choisie. L'élan fonctionne pour les deux anaphores lexicologiques (verbes réfléchis et réciproques) et pronoms.

L'exactitude mais ceci des états d'évaluation 75% doit être donnée une « bonification » pour ces résultats enjamber une assurance très large : l'évaluation a été basée sur une sélection aléatoire des genres, y compris des communiqués de presse, annonce de produits, articles, articles de magazine, et autre documente exister comme pages de World Wide Web.

3.4 Mitkov (2002)

Le principe de l'algorithme de R. Mitkov [13, Mitkov et autre] est de minimiser l'utilisation de données syntaxiques et sémantiques, qui sont assez coûteuses en termes de développement, pour la résolution d'anaphores. Le but est de faciliter l'implémentation tout en assurant un bon taux de réussite sur le traitement de manuels techniques, et de permettre l'adaptation d'une langue à l'autre. Cette approche ne nécessite ni analyse syntaxique ni analyse sémantique, mais prend simplement en entrée la sortie d'un étiqueteur morpho- syntaxique. L'algorithme consiste en l'application de simples heuristiques préférentielles (antecedent indicators) basées sur des données empiriques.

L'algorithme se déroule de la façon suivante :

Les groupes nominaux apparaissant au plus à une distance de deux phrases de l'anaphore à résoudre sont d'abord identifiés. Ensuite, l'accord des traits de genre et nombre entre chacun des groupes nominaux et l'anaphore est vérifié. C'est ici que les heuristiques sont appliquées en séquence pour calculer un score pour les groupes nominaux antécédents potentiels. A chacun des candidats est attribué un score (-1, 0, 1, ou 2) pour chaque

heuristique, qui contribue à une somme totale. Le candidat avec le meilleur score total est sélectionné comme antécédent.

Les heuristiques utilisées dans le processus sont en rapport avec la saillance, la répétition d'expressions, la distance référentielle, et la topologie lexicale du texte.

En voici l'inventaire :

– Definiteness : la définitude est un facteur qui favorise un antécédent. Les groupes nominaux définis sont des antécédents plus probables que les indéfinis. Les GN définis ont un score de 0 et les indéfinis sont pénalisés avec -1.

– Givenness : les GN dans les phrases précédentes qui représentent le thème de l'énoncé (« the given information ») sont considérés comme de bons candidats. L'heuristique utilisée pour déterminer le thème est que c'est le premier GN dans une phrase non-impérative. Cela est dû à l'observation que, dans un texte, le thème apparaît en premier, et crée un lien avec le texte précédent. Les GN thème ont un score de 1, les autres 0.

– Indicating verbs : si un verbe appartient à l'ensemble des verbes « indicateurs », le premier GN qui le suit est considéré comme le candidat préféré. L'ensemble de ces verbes est V = {discuss, present, illustrate, identify, summarise, examine, describe, define, show, check, develop, review, report, outline, consider, investigate, explore, assess, analyse, synthesise, study, survey, deal, cover}. Les observations montrent que les noms qui suivent ces verbes sont particulièrement saillants, et c'est cela qui fait que ces verbes sont de bons indicateurs.

– Lexical reiteration : la répétition lexicale attribue 2 points aux groupes nominaux qui sont répétés deux fois ou plus dans la même phrase, 1 point si répété une fois, 0 points sinon. Cela comprend les synonymes qui peuvent être précédés de pronoms démonstratifs (The cartridge...this cartridge), et les GN ayant la même tête lexicale (The toner bottle, the bottle of toner, the bottle).

– Section heading preference : si un GN apparaît dans le titre d'une section du document, alors il est favorisé d'un point, 0 sinon.

– « Non-prepositional » noun phrases : un groupe nominal qui ne fait pas partie d'un groupe prépositionnel (0 points) est préférable à un GN qui fait partie d'un GP (-1 point).

– Collocation pattern preference : ce critère favorise les GN qui apparaissent dans une construction identique au pronom. La préférence est limitée aux contextes de la forme « GN (pronom), verbe » et « verbe, GN (pronom) ».

Ex. Press the keyi down...press iti again.

– Immediate reference : cette préférence rend compte d'une construction fréquente dans les manuels techniques. Dans la suite « ...(You) Vl GN ...

CONJ (you) V2 it (CONJ (you) V3 it) » où CONJ représente une conjonction (and, or, before, after...), le GN qui suit Vl est un antécédent très probable pour le pronom it qui apparaît immédiatement après V2. On lui accorde 2 points, 0 aux autres GN.

– Referential distance : moins un GN est éloigné d'une anaphore, plus il est probable comme antécédent. Dans les phrases complexes, les GN dans la proposition précédente sont les meilleurs candidats (2 points), suivi des GN dans la phrase précédente (1), ensuite deux phrases avant (0) et finalement 3 phrases avant (-1). Pour les anaphores dans les phrases simples, les GN dans la phrase précédente ont le meilleur score (1 point), ensuite ceux qui se trouvent 2 phrases avant (0), et finalement 3 phrases avant (-1).

– Term preference : les GN qui appartiennent au domaine couvert par le texte sont favorisés (1 point) par rapport à ceux qui n'y appartiennent pas.

Les pronoms non anaphoriques, comme le it pléonastique, sont éliminés par un « filtre référentiel ». Les cataphores ne sont pas traitées par l'algorithme, qui se décrit de la façon suivante :

– Examiner les trois phrases précédant l'anaphore (si disponibles) et rechercher les GN à gauche de l'anaphore.

– Eliminer de la liste de candidats potentiels : ceux dont les traits de nombre et genre sont incohérents avec l'anaphore.

– Appliquer les heuristiques préférentielles et assigner des points. Le candidat avec le score total le plus élevé est l'antécédent. Si deux candidats ont le même score, celui avec le meilleur score pour «immediate reference » est

préféré. Si cette préférence n'aide pas à trancher, alors c'est « collocation pattern » qui décide; sinon « indicating verbs », sinon choisir le candidat le plus récent.

Les évaluations effectuées par l'auteur ont montré un taux de réussite assez favorable (89,7%) sur un corpus de manuels techniques. De plus, il a été adapté au polonais et à l'arabe avec des taux de réussite encore meilleurs (93,3% et 95,2% respectivement).

4. L'approche proposée

Après ce que nous avons vu dans les chapitres précédents, nous allons présenter une approche de résolution et concevoir une petite application apte à tester l'algorithme.

4.1 Architecture générale.

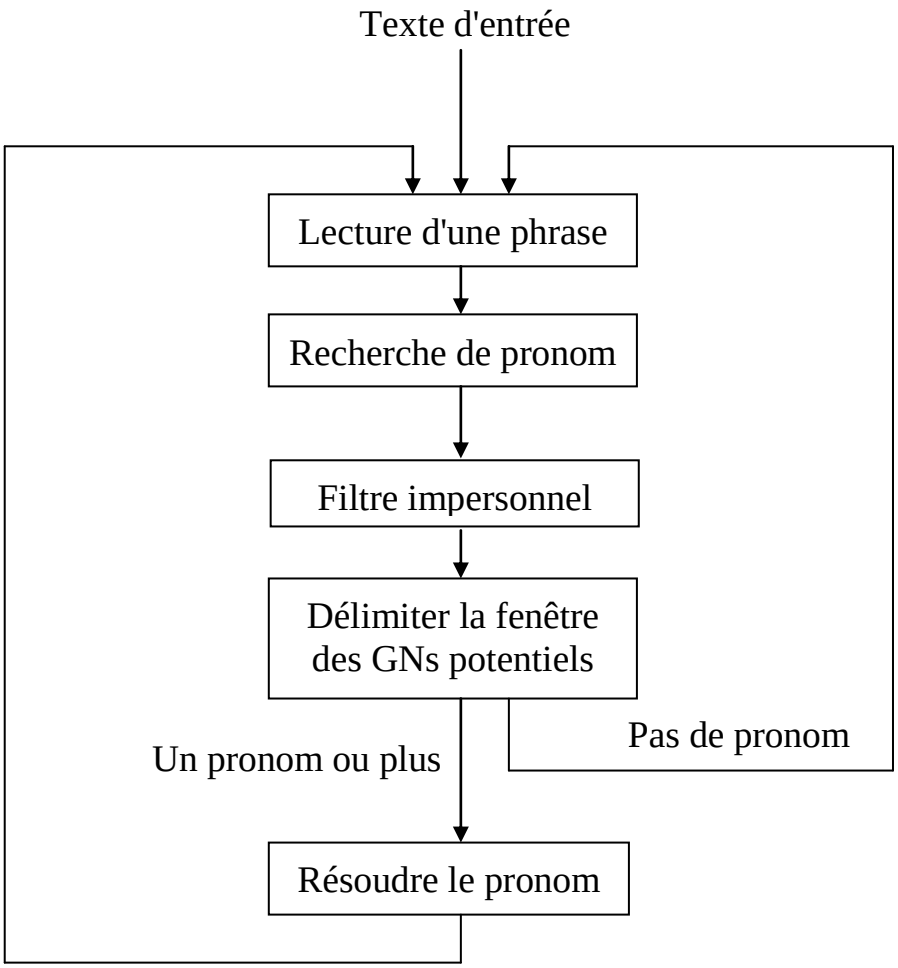


Figure: L'identification des éléments intervenant dans la résolution des

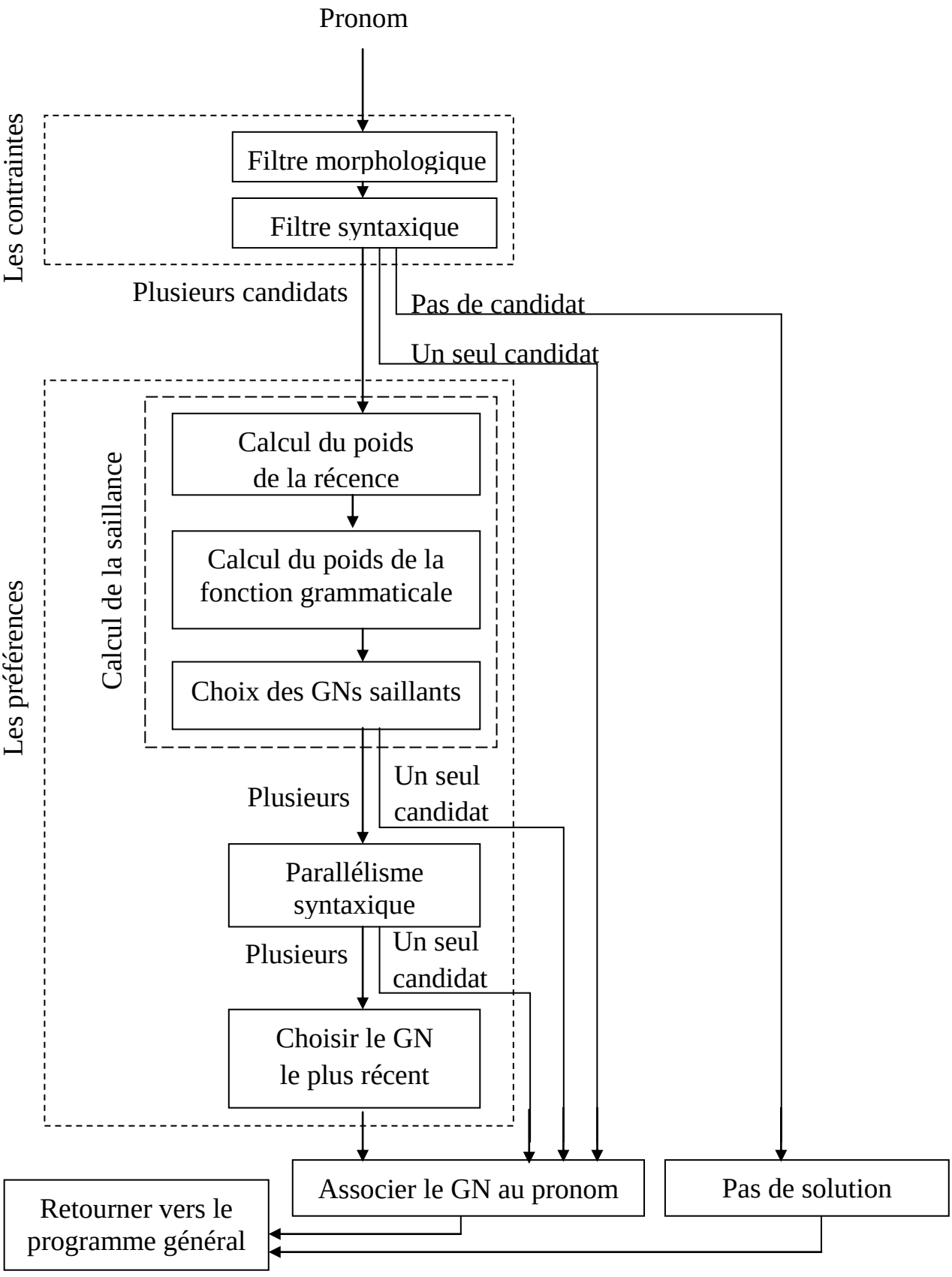


Figure: Le schéma de résolution des anaphores de l'algorithme PRAR

Donc le processus de la résolution des anaphores pronominales passe par les étapes suivantes:

1.Analyser la phrase en cours en recherchant d'identifier des pronoms.

2.Pour chacun pronom P, en vérifie s'il est personnel ou pas.

3.On construit un ensemble NPs des antécédents potentiels.

```
candidats (NPs,P,Discours):-
chercheP(Avant,P,Discour),
chercheNP(NPs,Avant).
```

4.Pour chaque paire (NPs, P), éliminer des antécédents en reposant sur un ensemble de contraintes,

```
conflit (L_candidats_en_accord,P,Discour):-
    candidats(Liste_candidats,P,Discour),
    accord(L_candidats_en_accord,P,Liste_candidats).
```

5.Si pour une paire (NPs, P), il reste plus d'un antécédent possible dans A, réduire A à un seul élément en fonction d'un ensemble de préférences.

```
resolution (NP,P,Discours):-
conflit (Liste,P,Discours),
not Liste= [X|[]],
best(Liste,NP,P).
```

```
resolution (NP,P,Discours):-
conflit ([NP|[]],P,Discours).
```

4.1.1 L'utilisation de la représentation syntaxique:

Le niveau de base d'analyse linguistique pour la résolution des anaphores est une partie de représentation syntaxique, enrichie d'annotations de fonction syntaxique. Cette entrée est générée manuellement. Par contre il existe des Taggers comme The Morphosyntactic Tagging System de Voutilainen[19, Voutilainen et autres], et celui de Karlsson [09,Karlsson]. Et des analyseurs syntaxiques comme LEOPAR de B. Guillaume et G. Perrier [i, Guillaume et autres], Mais, ils ne sont pas disponibles.

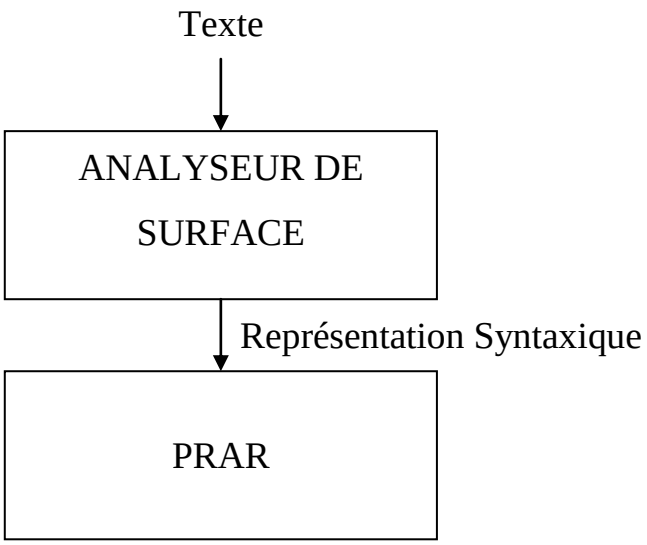


Figure: l'entée de l'algorithme PRAR

La représentation syntaxique fournit une représentation simple de la structure du texte: Pour chaque élément lexical dans une phrase, il fournit un ensemble de valeurs qui indiquent les caractéristiques lexicales, grammaticales et syntaxiques dans le contexte dans lequel il apparaît.

4.1.2 Une grammaire adéquate pour l'analyse

Dans notre étude, on a utilisé la grammaire suivante:

S→Gn,Gv

S→Gn,Pr

Gn→Art,Adj,Nom/Pronom

Gp→Prep,Gn

Gv→V,Gn/V,Gp/V,Adv,Gn/V,Adv,Gp

Adj→A11/ε f

Pr→Pronprop,S

Pronprop→que/qui

1.Le texte est un ensemble de phrases.

Discours= [Phrase]

Discours= [Phrase | Discours]

2.Une phrase contient un groupe nominal et un autre verbal.

Phrase= phrase(Gn,Gv)

3.Un groupe verbale contient un verbe, et peut être un groupe nominal ou/et une proposition

Gv= gv(V, Gn, Prep)

Gv= gv(V, Adv, Prep)

Gv= gv(V, Gn)

Gv= gv(V, Prep)

Gv= gv(V)

4.Un groupe nominal a une valeur (le texte) et enrichi avec des informations utiles, cette valeur peut être un pronom ou autre.

Gn=gn(Valnp,Genre,Nombre,Personne,GFUN, NPnbr, Saillance)

Valnp=Pronom

Valnp=Autre

5.Un pronom a une valeur (texte) et un certain nombre de propriétés

Pronom= pronom(Valp,Genre,Nombre,Personne,GFUN, Pnbr)

4.2 Elimination des pronoms non anaphoriques :

Dans un processus de résolution d'anaphores cette distinction est importante car elle permet d'éviter une tentative de résolution lorsqu'il n'y a pas d'antécédent.

- verbes météorologiques
- *ça/cela* barde, va, boume, roule
- *ça/cela* sujet de verbe à complétive
- constructions clivées
- *ce qui/que/dont/à quoi* - relative périphrastique sujet
- *ce qui/que/dont/à quoi* - relative périphrastique objet direct/indirect
- *est-ce que* – interrogatives
- *c'est..qui/que* - clivées déclaratives
- *est-ce..que/qui* - clivées interrogatives
- *à/de ce que* - objet prépositionnel indirect
- c'est + GA/GP + que

4.3 La recherche des pronoms

Le processus recherche les pronoms dans la représentation syntaxique d'entrée, quand il trouve un, il délimite le texte qui le précède.

- S'il trouve un pronom dans le groupe nominal sujet d'une phrase, alors l'antécédent peut être dans une autre phrase qui la précède ou il n'aura pas d'antécédent (cataphore).

[Elle] se réfère à l'antécédent. [L'anaphore] a une coréférence virtuelle avec son antécédent. (cataphore)

[L'anaphore] a une coréférence virtuelle avec son antécédent. [Elle] se réfère à l'antécédent. (Ici l'antécédent est dans la phrase précédente)

```
cherche P([],pronom(Val,Genre,Nombre,Personne,GFUN, Pnbr),Discours):-
Discours=[phrase(X,Y)|RestDiscours],
X=gn(pronom(Val,Genre,Nombre,Personne,GFUN,Pnbr),
Genre,Nombre,Personne,GFUN, Pnbr, Saillance).
```

- S'il trouve un pronom dans le groupe verbal d'une phrase, alors le groupe nominale de la même phrase peut être l'antécédent de ce pronom (dans les deux cas où le groupe verbale contient une proposition ou non)

[L'Algérie] a remporté [sa] liberté.

```
cherche P([phrase(X,gv(vid))],pronom(Val,Genre,Nombre,Personne,GFUN,
Pnbr),Discours):-
Discours=[phrase(X,Y)|RestDiscours],
Y=gv(V,gn(pronom(Val,Genre,Nombre,Personne,GFUN,Pnbr),
Genre,Nombre,
Personne,GFUN, Pnbr,Saillance)).
```

[L'anaphore] réfère à [son] antécédent.

```
chercheP([phrase(X,gv(vid))],pronom(Val,Genre,Nombre,Personne,GFUN,
Pnbr),Discour):-
Discours=[phrase(X,Y)|RestDiscours],
Y=gv(V,gn(pronom(Val,Genre,Nombre,Personne,GFUN,Pnbr),
Genre,Nombre,
Personne, GFUN,Pnbr,Saillance),Z).
```

- S'il trouve un pronom dans une phrase, alors son antécédent peut être dans cette phrase ou dans une autre phrase qui la précède.

```
chercheP([phrase(X,Y)|Avant],pronom(Val,Genre,Nombre,Personne,GFUN,
Pnbr),Discours):-
Discours=[phrase(X,Y)|RestDiscours],
chercheP(Avant,pronom(Val,Genre,Nombre,Personne,GFUN, Pnbr),
RestDiscours).
```

4.4 La recherche des antécédents

Après l'identification des pronoms et des phrases qui précèdent chacun d'eux. On passe à examiner ces textes pour trouver les groupes nominaux candidats.

- On parcourt le texte phrase par phrase. Si on trouve un groupe nominal - même dans un groupe verbal - on l'ajoute à la liste des candidats possibles.

```
chercheNP([gn(X,Genre,Nombre,Personne,GFUN, NPnbr,
Saillance)|Y],Discours):-
Discours=[phrase(gn(X,Genre,Nombre,Personne,GFUN, NPnbr,
Saillance),gv(V))|Reste],
chercheNP(Y,Reste).
```

```
chercheNP([gn(X1,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)
|[gn(X2,Genre2,Nombre2,Personne2,GFUN2, NPnbr2,
Saillance2)|Y]],Discour):-
Discours=[phrase(gn(X1,Genre,Nombre,Personne,GFUN,NPnbr,
Saillance),
gv(V,gn(X2,Genre2,Nombre2,Personne2,GFUN2,
NPnbr2,Saillance2))) |Reste],
chercheNP(Y,Reste).
```

```
chercheNP([gn(X,Genre,Nombre,Personne,GFUN, NPnbr,
 Saillance)|Y],Discour):-
Discours=[phrase(gn(X,Genre,Nombre,Personne,GFUN,NPnbr,
Saillance),
gv(V,prep(Prep)))|Reste],
chercheNP(Y,Reste).
```

```
chercheNP([gn(X1,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)
|[gn(X2,Genre2,Nombre2,Personne2,GFUN2, NPnbr2,
Saillance2)|Y]],Discours):-
Discour=[phrase(gn(X1,Genre,Nombre,Personne,GFUN,NPnbr,
Saillance),
gv(V,gn(X2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,
Saillance2),
Prep))|Reste],
chercheNP(Y,Reste).
```

```
chercheNP(Y,Discours):-
Discours=[Element|Reste],
not Element=phrase(gn(X,Genre,Nombre,Personne,GFUN,
NPnbr,Saillance),gv(V)),
      not
Element=phrase(gn(X1,Genre,Nombre,Personne,GFUN,NPnbr,Saillance),
gv(V,gn(X2,Genre2,Nombre2,Personne2,GFUN2, NPnbr2, Saillance2))),
      not
Element=phrase(gn(X1,Genre,Nombre,Personne,GFUN,NPnbr,Saillance),
gv(V,prep(Prep))),
      not
Element=phrase(gn(X1,Genre,Nombre,Personne,GFUN,NPnbr,Saillance),
gv(V,gn(X2,Genre2,Nombre2,Personne2,GFUN2, NPnbr2,Saillance2), Prep)),
chercheNP(Y,Reste).
```

4.5 Les contraintes

Après qu'on a déterminé les pronoms et extraire les syntagmes nominaux qui les précèdent. On va appliquer une liste de contraintes sur ces candidats pour éliminer ceux qui ne puissent pas être des référents à ces pronoms. Le reste des candidats vont constituer un ensemble de conflit.

```
candidats (Liste,P,Discours):-
chercheP(Avant,P,Discours),
chercheNP(Liste,Avant).
```

```
conflit (L_candidats_en_accord, P, Discours):-
candidats(Liste_candidats, P, Discours),
accord(L_candidats_en_accord, P, Liste_candidats).
```

Cette élimination est réalisée par un ensemble de filtres (morphologique et syntaxique)

4.5.1 Le filtre morphologique

Il consiste à vérifier l'accord en genre, nombre et personne des groupes nominaux avec un pronom. On distingue deux cas où le pronom aura soi un genre défini (masculin ou féminine) ou bien un genre indéfini.

- 1er cas le genre défini:

Pour un pronom P, on ne garde que les candidats qui s'accordent en genre, nombre et personne avec ce pronom.

```
accord (En_accord, P, Liste_candidats):-
P=pronom(Val_P,Genre,Nombre,Personne,Fonction2,Pnbr),
not Genre=i,
accord1(En_accord, P, Liste_candidats).
```

```
accord1 ([N|Suite_en_accord], P, Liste_candidats):-
Liste_candidats = [N|Suite],
P=pronom(Val_P,Genre,Nombre,Personne,Fonction2,Pnbr),
N=gn(Val_Np,Genre,Nombre,Personne,Fonction,NPnbr,Saillance),
accord1 (Suite_en_accord, P, Suite).

accord1 (Suite_en_accord, P, Liste_candidats):-
Liste_candidats = [N|Suite],
P=pronom(Val_P,Genrep,Nombrep,Personnep,Fonction2,Pnbr),
N=gn(Val_Np,Genrenp,Nombrenp,Personnenp,Fonction,NPnbr,
Saillance ),
not (Genrenp=Genrep,Nombrenp=Nombrep,Personnenp=Personnep),
accord1 (Suite_en_accord, P, Suite).
```

- 2eme cas le genre indéfini (l, d, les, des) :

Mais pour les pronoms de genre indéfini, on ne prend que le nombre et la personne en compte

```
accord (En_accord, P, Liste_candidats):-
P=pronom(Val_P,i,Nombre,Personne,Fonction2,Pnbr),
accord2(En_accord, P, Liste_candidats).
```

```
accord2 ([N|Suite_en_accord], P, Liste_candidats):-
Liste_candidats = [N|Suite],
P=pronom(Val_P,i,Nombre,Personne,Fonction2,Pnbr),
N=gn(Val_Np,Genrenp,Nombre,Personne,Fonction,NPnbr,Saillance),
accord2 (Suite_en_accord, P, Suite).

accord2 (Suite_en_accord, P, Liste_candidats):-
Liste_candidats = [N|Suite],
P=pronom(Val_P,i,Nombrep,Personnep,Fonction2,Pnbr),
N=gn(Val_Np,Genrenp,Nombrenp,Personnenp,Fonction,NPnbr,
Saillance ),
not (Nombrenp=Nombrep,Personnenp=Personnep),
accord2 (Suite_en_accord, P, Suite).
```

4.5.2 Le filtre syntaxique

D'autres contraintes syntaxiques sont :

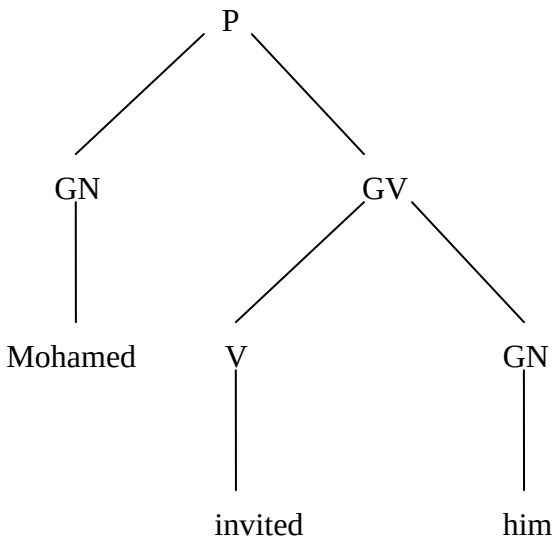
I. Un pronom non réfléchi et son antécédent ne peuvent pas apparaître dans la même phrase simple.

II. L'antécédent ne peut pas faire référence à une entité qui le c-commande.

C-commande:

La c-commande est définie chez Reinhart [16, Reinhart]comme : un nœud A est dit c-commander un nœud B, si A ne domine pas B, B ne domine pas A, et le premier nœud à ramifications (un nœud à ramifications est un nœud qui en domine immédiatement –au moins- deux autres au sein d'un arbre) qui domine A domine également B.

Exemple : Mohamed$_i$ invited him$_j$



Dans cet arbre:

Mouhamed c-commander him donc Mohamed ne pas être un antecedent du pronom him

4.6 Les préférences

Si l'ensemble des conflits contient un seul élément, alors il sera la solution du problème. Sinon, on doit sélectionner un élément de cette liste pour qu'il sera l'antécédent le plus probable. Cette sélection est dirigée par le calcule de la saillances. Mais s'il échoue, on choisi un selon le parallélisme syntaxique, et si le conflit reste en prend l'élément le plus proche du pronom.

```
calcule (Npscalcule,Npsnoncalcule,pronom
(Val_P,Genre,Nombre,Personne,Fonction,Pnbr)) :-
calculeGFUN(NPs_Gfun_calc,Npsnoncalcule) ,
recent(Npscalcule,NPs_Gfun_calc,Pnbr).
```

```
best(Liste,NPSP,P):-
calcule(Liste_calcule,Liste,P),
parallel_synt(NPP, Liste_calcule, P),
not NPP=[],
max(NPSP,NPP).
```

```
best(Liste,NPSP,P):-
calcule(Liste_calcule,Liste,P),
parallel_synt(NPP, Liste_calcule, P),
NPP=[],
max(NPSP,Liste_calcule).
```

```
max([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)|[]], L):-
L=[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)|
L2],
max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,
Saillance2)|NPS],L2),
Saillance > Saillance2.
```

```
max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,
Saillance2)|NPS], L):-
L=[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)
|L2],
max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,
Saillance2)|NPS],L2),
Saillance < Saillance2.
```

```
max([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,
Saillance)|[gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,
NPnbr2,Saillance2)|NPS]], L):-
L=[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,
Saillance)|L2],
max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,
NPnbr2,Saillance2)|NPS],L2),
Saillance = Saillance2.
```

4.6.1 Le calcul de saillance

Le calcul de la saillance est obtenu à partir des facteurs de saillances. Un référent de discours a un ou plusieurs facteurs de saillance. Chaque facteur de saillance aura un poids. La valeur de la saillance d'un candidat est la somme des valeurs des facteurs de saillance qu'il possède.

Les poids de chacun des types de facteur de saillance sont indiqués dans le tableau suivant. Noter que la saillance relative de certains de ces facteurs réalise une hiérarchie des rôles grammaticaux.

Type de facteur	Poids initial
Récence de phrase	100
Emphase soumise	80
Emphase existentielle	70
Emphase d'accusatif	50
Objet indirect et emphase oblique de complément	40
Emphase de nom principal	80
emphase Non - adverbale	50

Tableau1: Les types de facteur de saillances et leurs poids

Selon l'hiérarchie grammaticale de rôle, les règles d'évaluation assignent des poids plus élevés de saillance au

- sujet au-dessus du NPs non - sujet,
- les objets directs au-dessus d'autres compléments,
- les arguments d'un verbe au-dessus des adjonctions et les objets des phrases prépositionnelles (PP) adverbaux, et
- les noms principaux après des compléments des noms principaux.

4.6.2 La récence de phrase

Un facteur de saillance de récence de la phrase est créé pour la phrase courante. Sa portée est tous les référents de discours présentés par la phrase courante. La dégradation du facteur de saillance se produit quand en passe à une phrase précédente dans le texte.

Le tableau ci-dessous fournit le positionnement global des anaphores par rapport aux référents. On note que plus d'une anaphore sur 10 trouve son

référent dans une phrase qui n'est ni la phrase courante ni la précédente et qu'environ deux anaphores sur trois ont leur référent dans la phrase courante. [12, Laurent]

	Nombre	%
Référent 25 phrases avant l'anaphore	1	0,04 %
Référent 18 phrases avant l'anaphore	1	0,04 %
Référent 17 phrases avant l'anaphore	1	0,04 %
Référent 16 phrases avant l'anaphore	2	0,08 %
Référent 15 phrases avant l'anaphore	1	0,04 %
Référent 14 phrases avant l'anaphore	1	0,04 %
Référent 13 phrases avant l'anaphore	1	0,04 %
Référent 12 phrases avant l'anaphore	4	0,16 %
Référent 11 phrases avant l'anaphore	2	0,08 %
Référent 10 phrases avant l'anaphore	3	0,11 %
Référent 9 phrases avant l'anaphore	2	0,08 %
Référent 8 phrases avant l'anaphore	4	0,16 %
Référent 7 phrases avant l'anaphore	6	0,24 %
Référent 6 phrases avant l'anaphore	11	0,44 %
Référent 5 phrases avant l'anaphore	13	0,52 %
Référent 4 phrases avant l'anaphore	27	1,08 %
Référent 3 phrases avant l'anaphore	51	2,03 %
Référent 2 phrases avant l'anaphore	129	5,14 %
Référent phrase précédant l'anaphore	563	22,43 %
Référent même phrase que l'anaphore	1679	66,89 %
Référent phrase suivant l'anaphore	7	0,28 %
Total	2510	100 %

Tableau2: le pourcentage de la distance entre l'anaphore et son antécédent

- Si le groupe nominal et le pronom sont dans la même phrase. La valeur donnée à la récence est 100, et si le groupe nominal est dans

une phrase précédente on diminue la valeur par 10 à chaque retour en arrière.

```
recent([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance2)|Restacal
culer],
[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1) | Rest], Pnbr) :-
Pnbr < NPnbr+10,
recent (Restacalculer, Rest, Pnbr),
Saillance2= Saillance1+(10+NPnbr- Pnbr)*10.
```

- Si on arrive à un groupe nominal de dix 10 phrases avant l'anaphore ou plus, la valeur de la récence sera supprimée (ou égale à 0).

```
recent([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1)|Restacal
culer],
[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1) | Rest], Pnbr):-
                Pnbr >= NPnbr+10,
                recent (Restacalculer, Rest, Pnbr).
```

4.6.3 Utilisation dans un sujet

Si le groupe nominal a une fonction grammaticale d'un sujet. Il aura une valeur de 80.

```
calculeGFUN([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,
Saillance2)|Restcalcule], [gn(Val_Np,Genre,Nombre,Personne,
GFUN,NPnbr,Saillance1)| Rest]) :-
                GFUN= subject,
                Saillance2= Saillance1+80,
                calculeGFUN (Restcalcule, Rest).
```

4.6.4 Utilisation dans une expression existentielle:

Prédicat nominal dans une construction existentielle (« There is...») est saillant. (70 de valeur de saillance).

4.6.5 Utilisation dans une expression accusative :

Le candidat en position d'objet direct (complément verbal dans le cas d'accusatif) est saillant, mais pas autant qu'un sujet. (50 de valeur de saillance).

```
calculeGFUN ([gn(Val_Np,Genre,Nombre,Personne,GFUN,
NPnbr,Saillance2) | Restcalcule],[gn(Val_Np,Genre,Nombre,
Personne,GFUN,NPnbr,Saillance1)| Rest]) :-
                GFUN= directobject,
                Saillance2= Saillance1+50,
                calculeGFUN (Restcalcule, Rest).
```

4.6.6 Utilisation dans une expression d'objet indirect

Un objet indirect est moins saillant qu'un élément objet direct. (40 de valeur de saillance).

```
calculeGFUN ([gn(Val_Np,Genre,Nombre,Personne,GFUN, NPnbr,Saillance2) |
Restcalcule],
                [gn(Val_Np,Genre,Nombre,Personne,GFUN,
NPnbr,Saillance1)| Rest]) :-
                GFUN= indirectobject,
                Saillance2= Saillance1+40,
                calculeGFUN (Restcalcule, Rest).
```

4.6.7 Apparence de nom principal :

Une tête nominale dans un groupe nominal complexe est plus saillante qu'un nom non tête, qui est pénalisé. Ce facteur augmente la valeur de saillance du NP qui n'est pas enfoncée dans l'autre NP (en tant que son complément ou adjonction).

4.6.8 Utilisation dans une expression Non - adverbale :

Ce facteur pénalise les GN dans des constructions adverbiales.

4.6.9 Le parallélisme syntaxique

Le parallélisme syntactique pourrait être tout à fait utile quand d'autres contraintes ou préférences ne sont pas en mesure pour proposer un antécédent non ambigu. Cette préférence est donnée aux groupes nominaux avec la même fonction syntactique que l'anaphore.

```
parallel_synt ([], [], P).
```

```
parallel_synt ([NP|NPSUITE], LESNP, P):-
        LESNP = [NP|Suite],
NP=gn(Val_Np,Genre,Nombre,Personne,Fonction,NPnbr,Saillance),
        P=pronom(Val_P,Genre2,Nombre,Personne,Fonction,Pnbr),
        parallel_synt (NPSUITE,Suite, P).
```

```
parallel_synt (NPSUITE, LESNPS, P):-
        LESNPS = [NP|Suite],
        NP=gn(Val_Np,Genre,Nombre,Personne,Fonction1,NPnbr,Saillance),
        P=pronom(Val_P,Genre2,Nombre,Personne,Fonction2,Pnbr),
        not Fonction1= Fonction2,
        parallel_synt (NPSUITE, Suite, P).
```

4.6.10 Le parallélisme sémantique

Il n’a pas été utilisé. Il est plus puissant que le parallélisme syntaxique mais il nécessite un système qui peut identifier les rôles sémantiques. Il favorise les groupes nominaux qui ont le même rôle sémantique que l'anaphore.

5. Test et validation

Dans cette partie on va tester l'algorithme avec quelques exemples. Pour implémenter cette approche on a utilisé Prolog Inference Engine qui est fourni avec visuel prolog 6.

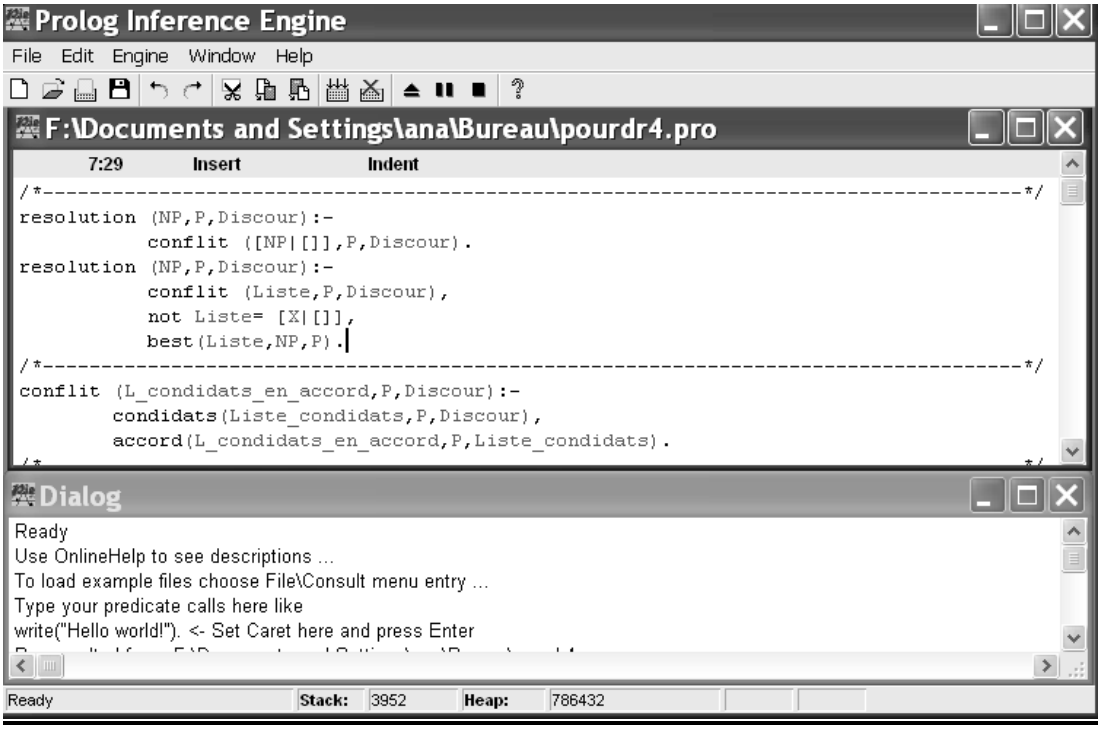
C'est une application MDI. Il y a deux types de fenêtres sur le bureau PIE:

- L'editeur: permet à l'utilisateur d'entrer et de modifier le code Prolog;
- La fenêtre de dialogue: Représente la console du Prolog et elle permet à l'utilisateur de contrôler le système du Prolog.

On a choisis Prolog car il est l'un des principaux langages de programmation logique. son nom est un acronyme de PROgrammation en LOGique. Il a été créé par Alain Colmerauer et Philippe Roussel vers 1972. Le but était de créer un langage de programmation qui permettrait d'utiliser l'expressivité de la logique.

Prolog est utilisé dans de nombreux programmes d'intelligence artificielle et dans le traitement de la linguistique par ordinateur comme notre cas (surtout ceux concernant les langages naturels). Sa syntaxe et sa sémantique sont considérées comme très simples et claires.

Prolog est basé sur le calcul des prédicats du premier ordre ; cependant il est restreint dans sa version initiale à n'accepter que les clauses de Horn. Les concepts fondamentaux sont l'unification, la récursivité et le retour sur trace.



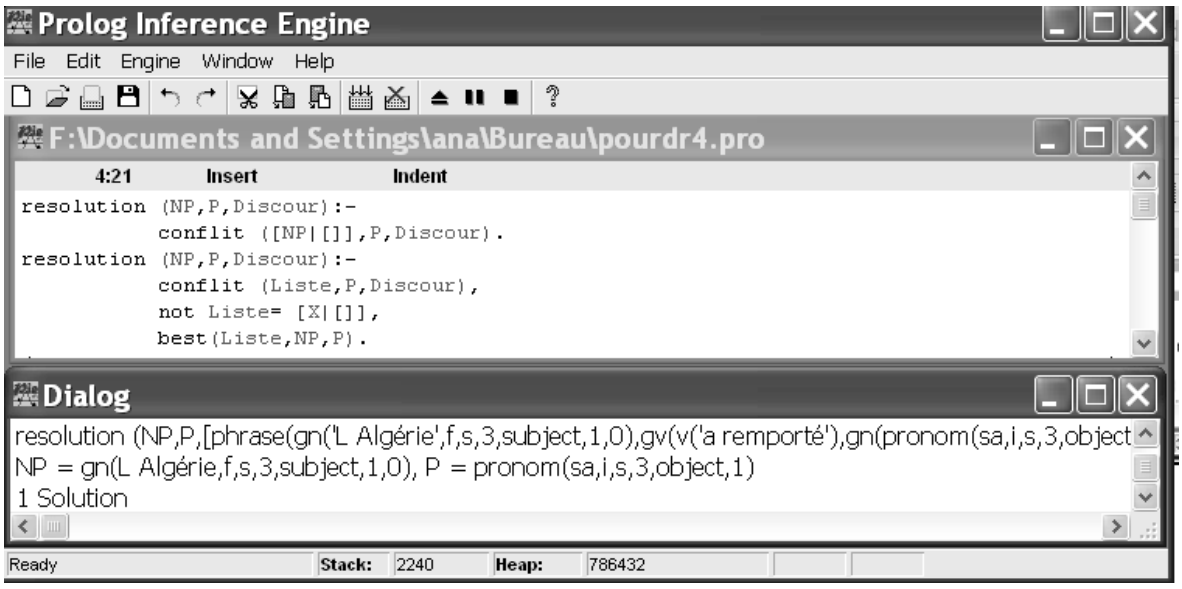
Test 1

Le discours:

L'Algérie a remporté sa liberté.

Le texte d'entrée:

resolution (NP,P,[phrase(gn('L Algérie',f,s,3,subject,1,0),gv(v('a remporté'),gn(pronom(sa,i,s,3,object,1),i,s,3,object,1,0), prep(liberté)))])



Résultat:

NP = gn(L Algérie,f,s,3,subject,1,0),

P = pronom(sa,i,s,3,object,2)

1 Solution

Discussion:

Le processus a trouvé un seul pronom P = pronom(sa,i,s,3,object,2) qui occupe une position de complément d'objet direct dans la phrase. Et pour ce pronom il a trouvé l'antécédent qui est le groupe nominale 'L'algerie'.

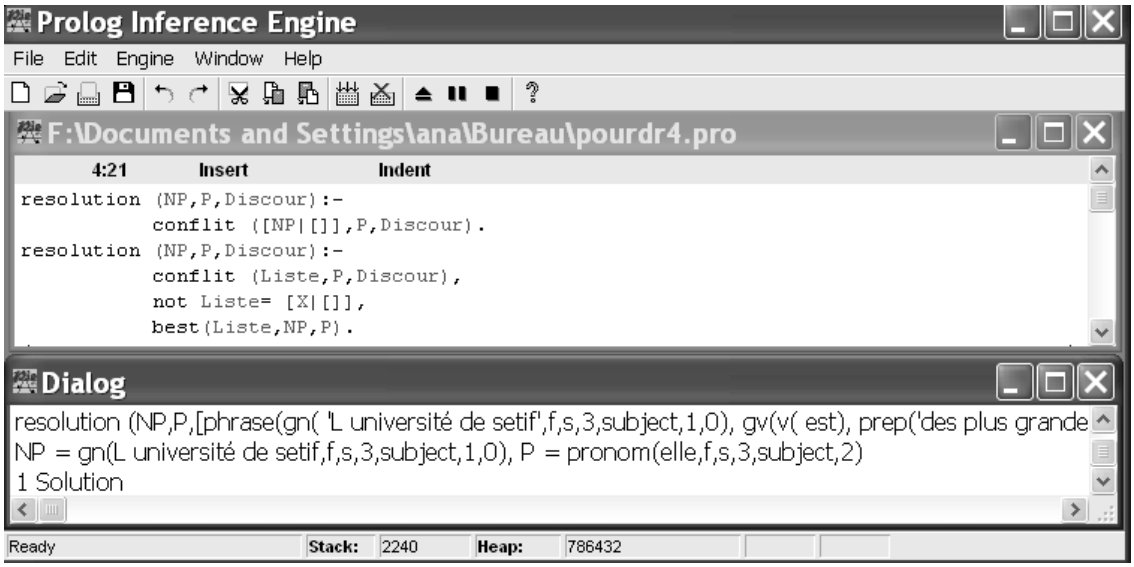
Test 2

Le discours:

L'université de setif est des plus grandes universités algériennes. Elle enseigne la plus parts des disciplines.

Le texte d'entrée:

resolution (NP,P,[phrase(gn('L université de setif',f,s,3,subject,1,0), gv(v(est), prep('des plus grandes universités algériennes'))),phrase(gn(pronom(elle,f,s,3,subject,2),f,s,3,subject,2,0),gv(v(enseigne), prep('la plus parts des disciplines')))])



Résultat:

NP = gn(L université de setif,f,s,3,subject,1,0),

P = pronom(elle,f,s,3,subject,2)

1 Solution

Discussion:

Le processus a trouvé un seul pronom P = pronom(elle,f,s,3,subject,2)

féminin singulier qui occupe une position de sujet dans la deuxième phrase.et pour ce pronom il a choisis le groupe nominale "L'université de setif" comme antécédent. Ce groupe est dans la phrase précédente.

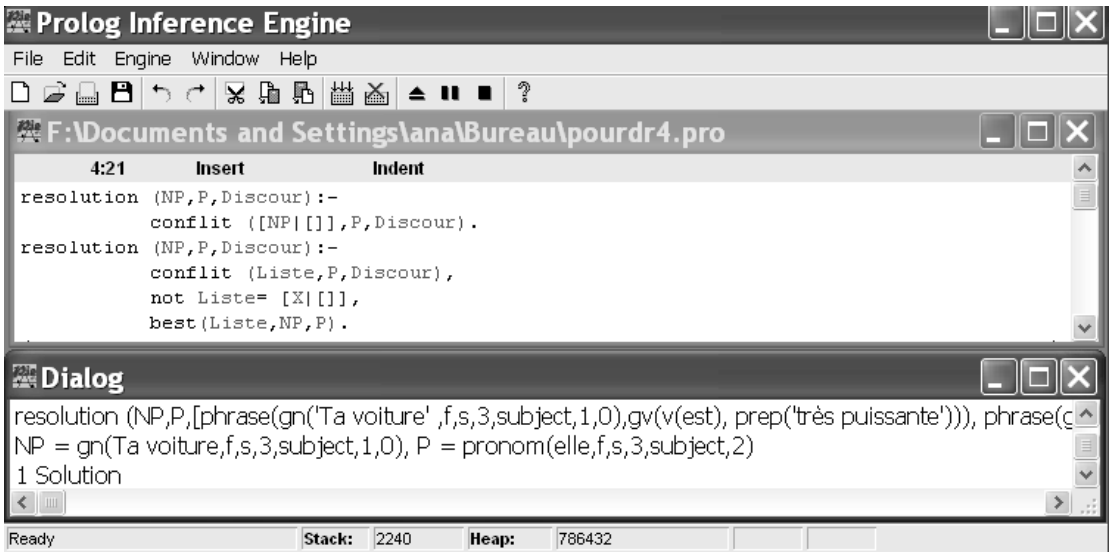
Test 3

Le discours:

Ta voiture est très puissante. Elle dépasse cent kilomètre à l'heur dans cinq secondes.

Le texte d'entrée:

resolution (NP,P,[phrase(gn('Ta voiture' ,f,s,3,subject,1,0),gv(v(est), prep('très puissante'))), phrase(gn(pronom(elle,f,s,3,subject,2),f,s,3,subject,2,0),gv(v(dépasse), prep('cent kilomètre à l heur dans cinq secondes')))])



Résultat:

NP = gn(Ta voiture,f,s,3,subject,1,0),

P = pronom(elle,f,s,3,subject,2)

1 Solution

Discussion:

Le processus a trouvé un seul pronom P = pronom(elle,f,s,3,subject,2) féminin singulier qui occupe une position de sujet dans la deuxième phrase. Et pour ce pronom il a choisis le groupe nominal " Ta voiture " comme antécédent.

Test 4

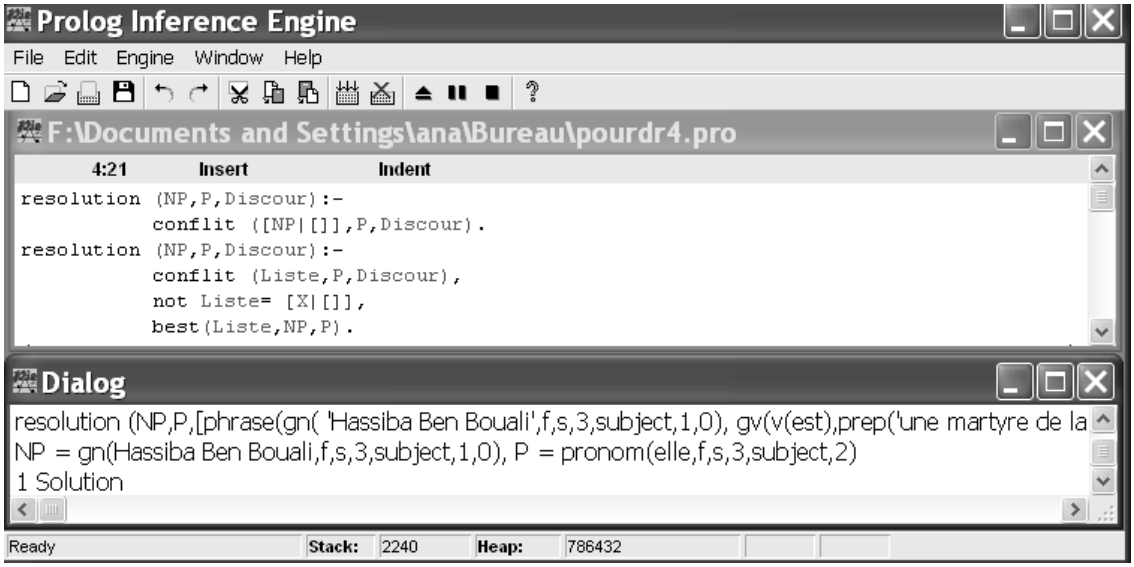
Le discours:

Hassiba Ben Bouali est une martyre de la révolution algérienne contre le colonialisme français. Elle était une militante et résistante durant la guerre d'Algérie,

Le texte d'entrée:

resolution (NP,P,[phrase(gn('Hassiba Ben Bouali',f,s,3,subject,1,0), gv(v(est),prep('une martyre de la révolution algérienne contre le colonialisme français'))),

phrase(gn(pronom(elle,f,s,3,subject,2),f,s,3,subject,2,0),gv(v(était), prep(' une militante et résistante durant la guerre d Algérie')))])



Résultat:

NP = gn(Hassiba Ben Bouali,f,s,3,subject,1,0),

P = pronom(elle,f,s,3,subject,2)

1 Solution

Discussion:

Le processus a trouvé un seul pronom P = pronom(elle,f,s,3,subject,2) féminin singulier qui occupe une position de sujet dans la deuxième phrase. Et pour ce pronom il a choisis le groupe nominale " Hassiba Ben Bouali " comme antécédent.

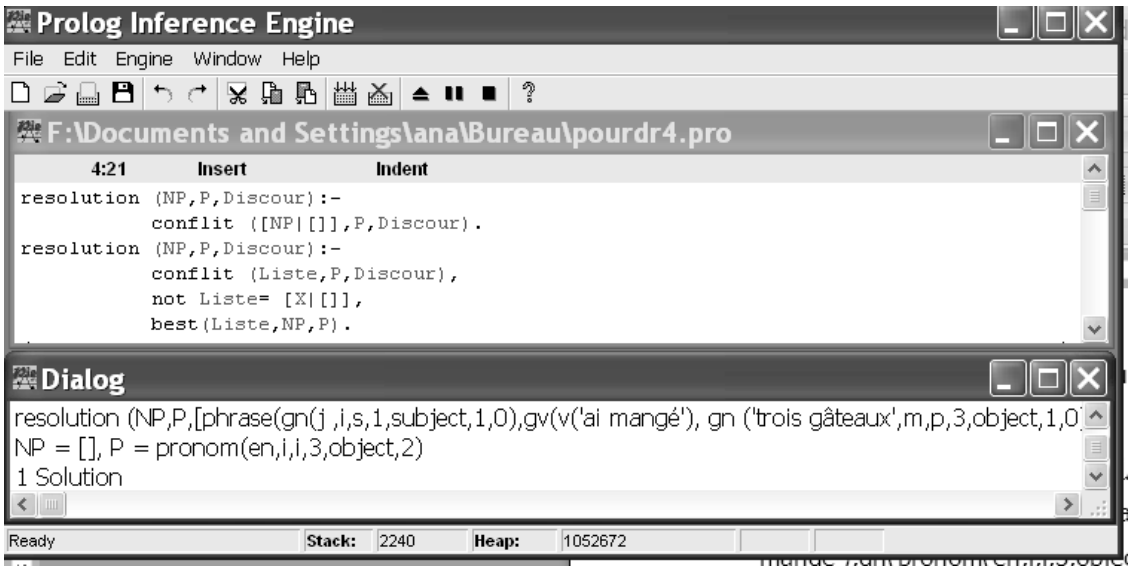
Test 5

Le discours:

J'ai mangé trois gâteaux et toi tu en as mangé cinq.

Le texte d'entrée:

resolution (NP,P,[phrase(gn(j ,i,s,1,subject,1,0),gv(v('ai mangé'), gn ('trois gâteaux',m,p,3,object,1,0))),phrase(gn(tu ,i,s,2,subject,2,0),gv(v('as mangé'),gn(pronom(en,i,i,3,object,2),i,i,3,object,2,0), prep(' cinq'))])



Résultat:

NP = [], P = pronom(en,i,i,3,object,2)

1 Solution

Discussion:

Le processus a trouvé un seul pronom P = pronom(en,i,i,3,object,2) qui occupe une position de complément d'objet direct dans la deuxième phrase. Et pour ce pronom il n'a pas trouvé un antécédent.

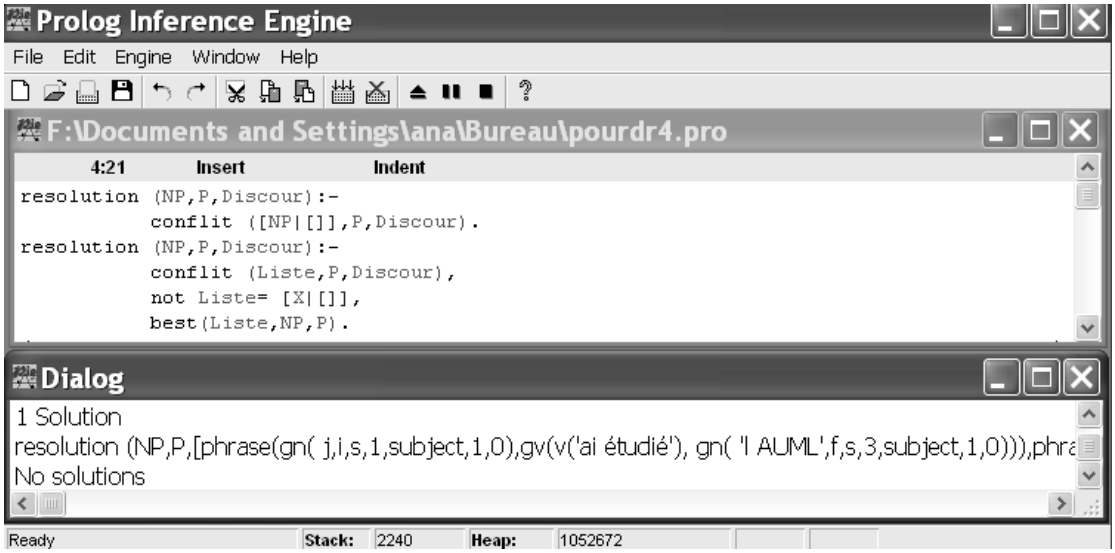
Test 6

Le discours:

J'ai étudié l'AUML. C'est une version de l'UML orienté agent.

Le texte d'entrée:

resolution (NP,P,[phrase(gn(j,i,s,1,subject,1,0),gv(v('ai étudié'), gn('l AUML',f,s,3,subject,1,0))),phrase(gn(pronom(c,i,s,3,subject,2),i,i,3,subject,2,0),gv(v(est) ,gn(' une version de l UML',f,s,3,object,2,0) , prep('orienté agent'))])



Résultat:

No solutions

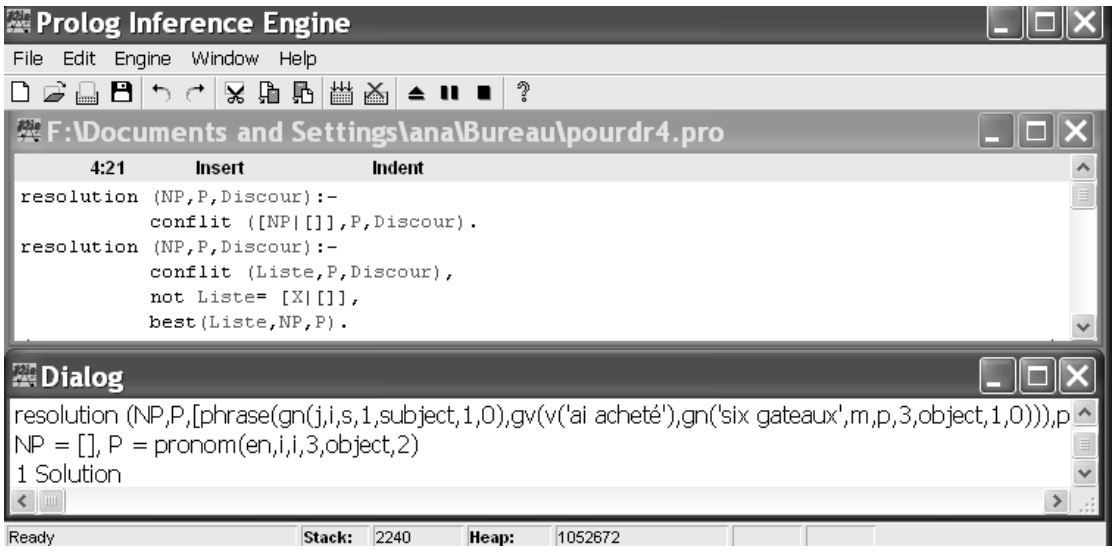
Test 7

Le discours:

J'ai acheté six gâteaux, j'en ai mangé deux sur place.

Le texte d'entrée:

resolution (NP,P,[phrase(gn(j,i,s,1,subject,1,0),gv(v('ai acheté'),gn('six gateau x',m,p,3,object,1,0))),phrase(gn(j,i,s,1,subject,1,0),gv(v('ai mangé'),gn(prono m(en,i,i,3,object,2),i,i,3,object,2,0), prep(' sur place')))])



Résultat:

NP = [], P = pronom(en,i,i,3,object,2)

1 Solution

Discussion:

Le processus a trouvé un seul pronom P = pronom(en,i,i,3,object,2) qui occupe une position de complément d'objet direct dans la deuxième phrase. Et pour ce pronom il n'a pas trouvé un antécédent.

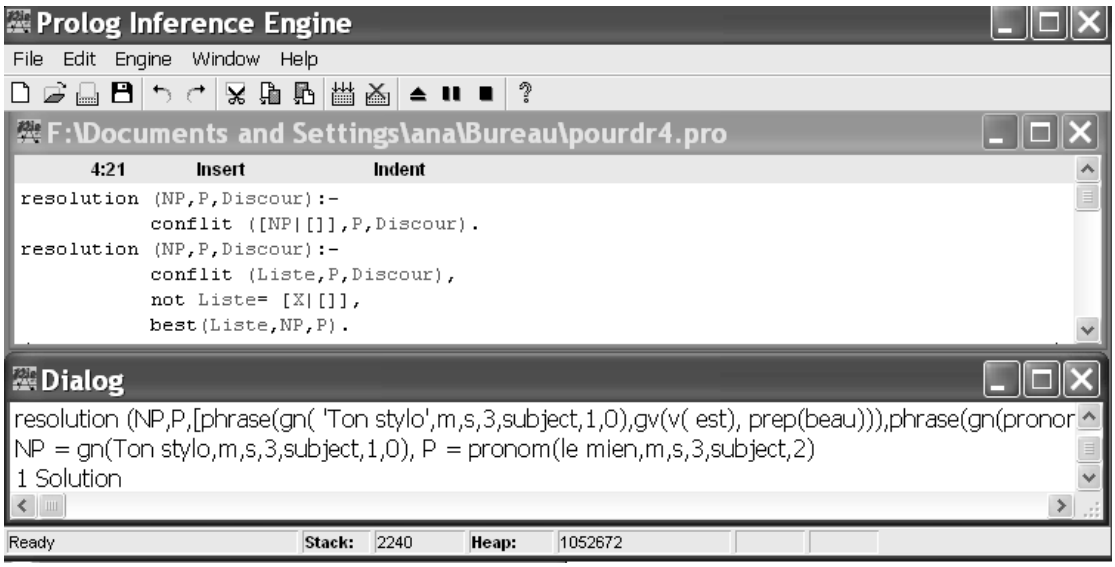
Test 8

Le discours:

Ton stylo est beau mais le mien écrit mieux.

Le texte d'entrée:

resolution (NP,P,[phrase(gn('Ton stylo',m,s,3,subject,1,0),gv(v(est), prep(beau))),phrase(gn(pronom('le mien',m,s,3,subject,2),m,s,3,subject,2,0),gv(v(écrit), prep(mieux)))])



Résultat:

NP = gn(Ton stylo,m,s,3,subject,1,0),

P = pronom(le mien,m,s,3,subject,2)

1 Solution

Discussion:

Le processus a trouvé un seul pronom P = pronom(le mien,m,s,3,subject,2) masculin singulier qui occupe une position de sujet dans la deuxième phrase. Et pour ce pronom il a choisis le groupe nominale " Ton stylo " comme antécédent.

Test 9

Le discours:

L'anaphore a une coréférence virtuelle avec son antécédent. Elle se réfère à l'antécédent

Le texte d'entrée:

resolution (NP,P,[phrase(gn('L anaphore',f,s,3,subject,1,0),gv(v(a),gn(' une coréférence virtuelle ',f,s,3,object,1,0), prep(' avec son antécédent'))), phrase(gn(pronom(elle,f,s,3,subject,2),f,s,3,subject,2,0),gv(v(réfère),gn(pronom(se,i,s,3,object,2),i,s,3,object,2,0), prep(' à l antécédent')))])

Résultat:

No solutions

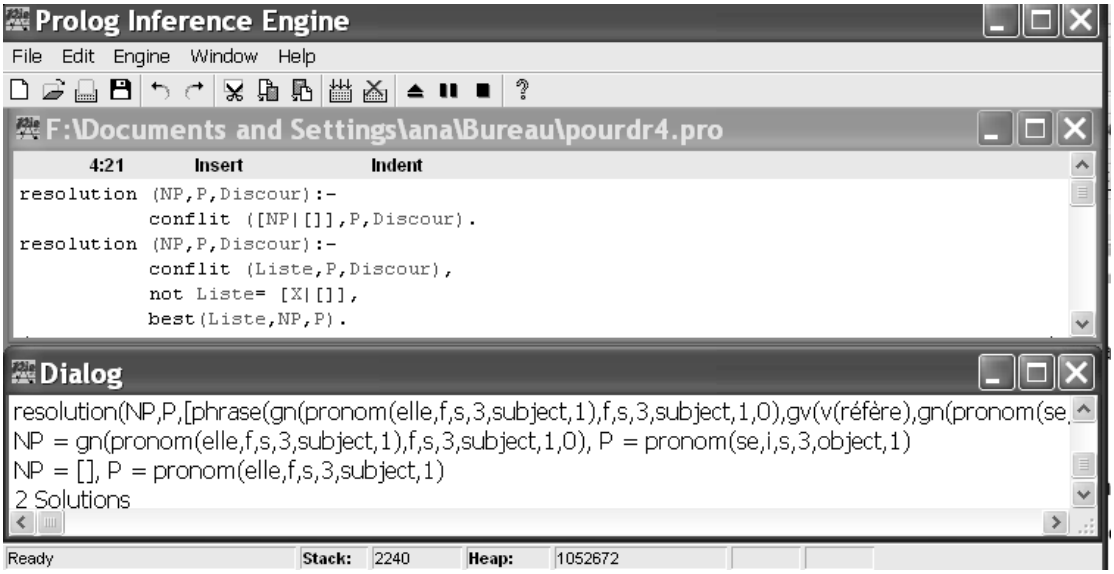
Test 10

Le discours:

Elle se réfère à l'antécédent. L'anaphore a une coréférence virtuelle avec son antécédent.

Le texte d'entrée:

resolution(NP,P,[phrase(gn(pronom(elle,f,s,3,subject,1),f,s,3,subject,1,0),gv(v(réfère),gn(pronom(se,i,s,3,object,1),i,s,3,object,1,0), prep(' à l antécédent'))),phrase(gn('L anaphore',f,s,3,subject,1,0),gv(v(a),gn(' une coréférence virtuelle ',f,s,3,object,2,0), prep(' avec son antécédent')))])



Résultat:

NP = gn(pronom(elle,f,s,3,subject,1),f,s,3,subject,1,0),

P = pronom(se,i,s,3,object,1)

NP = [],

P = pronom(elle,f,s,3,subject,1)

2 Solutions

Discussion:

Le processus a trouvé deux pronoms

P1= pronom(se,i,s,3,object,2) le pronom occupe une position de complément d'objet direct dans la deuxième phrase. Il se réfère au pronom personnelle elle sujet de la même.

Et le pronom P2= pronom(elle,f,s,3,subject,1) qui est cataphorique.

6. Conclusion et perspectives

Pour cette étude, on a voulu se placer dans l'optique d'une implémentation d'un programme de TAL, en exploitant des données de nature syntaxique et lexicale.

La modélisation qu'on propose fournit un cadre opératoire pour le traitement des anaphores pronominales. On a étudié un ensemble de relations rhétoriques adaptées au dialogue finalisé, et fourni les critères calculables pour l'inférence de ces relations. On a adapté l'algorithme de Lappin et Leass sur les points suivants:

- On examine le texte pour l'extraction des pronoms et les groupes nominaux qui les précèdent.

Θn élimine les référents du discours pour lesquelles une expression anaphorique ne peut se référer.

On sélectionne l'antécédent optimal parmi les candidats selon des préférences basées sur des concepts linguistique .

Il faut cependant prendre en compte la complexité des textes du corpus, qui pourrait grever les résultats. Il n'est donc pas absurde d'espérer que sur des textes plus simples, plus classiques, les résultats seront assez similaires.

Une première conclusion que l'on peut tirer de cette étude est que la syntaxe impose un grand nombre de contraintes rigides sur la sélection d'un antécédent. Certaines propriétés morpho-syntaxiques peuvent déterminer la relation anaphorique. De plus, une analyse d'autres facteurs pourrait être utile pour améliorer et optimiser le processus .

L'algorithme qu'on a formulé est fondé sur une grammaire simplifiée et il est efficace et robuste : après avoir déterminé les antécédents candidats, il évalue la relation qui existe entre une anaphore donnée et chacun de ces candidats. Par contre, il ne prend que très peu en compte la relation qui peut exister entre deux candidats. On peut considérer qu'il relève d'une approche impérative assez statique et qu'il ne vise pas à représenter la façon dont les humains interprètent un discours. De plus, il consiste à effectuer des calculs pour chaque occurrence d'un pronom, ce qui est n'est pas très optimisé.

Parmi les perspectives futures pour complété et amélioré l'algorithme que nous avons proposé on peut citer :

- inclure d'autres connaissances dans la résolution : sémantiques
- l'utiliser avec un Tagger efficace.
- Approfondir les contraintes : parallélisme syntaxique et les c-commandes, l'appliquer à d'autres types d'anaphores
- inclure les pronoms possessifs
- inclure les groupes nominaux de la forme *l'un d'eux*

7. Bibliographie

[01, Bittar] André Bittar «Un algorithme pour la résolution d'anaphores Événementielles ». *Master de Linguistique informatique* Juin 2006.

[02, Carter] D. Carter - Interpreting anaphora in natural language texts. Chichester: Ellis Horwood, 1987

[03, Carter 90] David M. Carter - Control issues in anaphor resolution. Journal of Semantics, 7, 1990

[04, Gaizauskas et autres] Gaizauskas, R., Hepple,M. and Huyck, C. (1998). A Scheme for Comparative Evaluation of Diverse Parsing Systems.

[05, Guillaume et autres] B. Guillaume & G. Perrier : LEOPAR, un analyseur syntaxique pour les grammaires d'interaction

[06, Halliday et autres] Halliday, M.A.K. et Hasan, R. (1976), « Cohesion in English», Londres, Longman.

[07, Hobbs] Hobbs, J. 1978. « Resolving pronoun references». Lingua, 44, 311-338.

[08, Kameyama] Kameyama M. « Anaphora in Natural Language Understanding ». Springer-Verlag, Berlin.

[09, Karlsson] Karlsson Fred. 1990. Constraint Grammar as a Framework for Parsing Unrestricted Text. H. Karlgren, ed., Proceedings of the 13th International Conference of Computational Linguistics, Vol. 3. Helsinki 1990, 168-173.

[10, Kennedy et autres] Christopher Kennedy, Branimir Boguraev « Anaphora for Everyone: Pronominal Anaphora Resolution without a Parser ». In: Proceedings of the 16th conference on Computational Linguistics, August 05-09, 1996, Copenhagen, Denmark.

[11, Lappin et autres] Shalom Lappin, Herbert J. Leass « An Algorithm for Pronominal Anaphora Resolution » computational Linguistics, 20(4):535-561

[12, Laurent] Dominique Laurent « De la résolution des anaphors » Published in 4th Discourse Anaphora and Anaphor Resolution Colloquium, Portugal (2002)".

[13, Mitkov et autres] Ruslan Mitkov, Richard Evans, Constantin Orasan(2002): « A New, Fully Automatic Version of Mitkov's Knowledge-Poor Pronoun Resolution Method ». In Proceedings of CICLing-2000, Mexico City, Mexico.

[14, Mitkov] Ruslan Mitkov « Factors in anaphora resolution: they are not the only things that matter. A case study based on two different approaches ». In Proceedings of the ACL'97/EACL'97 workshop on Operational factors in practical, robust anaphora resolution, 14-21. Madrid, Spain.

[15, Mitkov, 1999] Ruslan Mitkov « Anaphora resolution: The state of the art». Working paper, (Based on the COLING'98/ACL'98 tutorial on anaphora resolution), University of Wolverhampton, Wolverhampton.

[16, Reinhart] Reinhart T (1981) definit NP anaphora and c-command, Linguistic Inquiry 12,605-635

[17, Srinivas et autres] Srinivas, B. and Baldwin, B. 1996. Exploiting supertag representation for fast coreference resolution. In International Conference on NLP+IA/TAL+AI, Moncton, NB, Canada.

[18, Voutilainen et autres] Atro Voutilainen, Juha Heikkil¨a, and Arto Antilla. A constraint grammar of english: A performance-oriented approach. University of Helsinki, Department of General Linguistics, Publication No. 21, Hallituskatu 11– 13, SF-00100 Helsinki, Finland, 1992.

[19, Zribi-Hertz , 1996] Anne Zribi-Hertz. « L'anaphore et les pronoms: une introduction à la syntaxe générative »

8. Annexe

Description de l'algorithme en prolog

```
/*-------------------------------------------------------------------------------*/
resolution (NP,P,Discour):-
        conflit ([NP|[]],P,Discour).
resolution (NP,P,Discour):-
        conflit (Liste,P,Discour),
        not Liste= [X|[]],
        best(Liste,NP,P).
/*-------------------------------------------------------------------------------*/
conflit (L_condidats_en_accord,P,Discour):-
      condidats(Liste_condidats,P,Discour),
      accord(L_condidats_en_accord,P,Liste_condidats).
/*-------------------------------------------------------------------------------*/
condidats (Liste,P,Discour):-
      chercheP(Avant,P,Discour),
      chercheNP(Liste,Avant).
/*-------------------------------------------------------------------------------*/
chercheP([],pronom(Val,Genre,Nombre,Personne,GFUN, Pnbr),Discour):-
      Discour=[phrase(X,Y)|RestDiscour],
      X=gn(pronom(Val,Genre,Nombre,Personne,GFUN, Pnbr),Genre,Nombre,
Personne,GFUN, Pnbr, Saillance).

chercheP([phrase(X,gv(vid))],pronom(Val,Genre,Nombre,Personne,GFUN, Pnb
r),Discour):-
      Discour=[phrase(X,Y)|RestDiscour],
      Y=gv(V,gn(pronom(Val,Genre,Nombre,Personne,GFUN, Pnbr),Genre,No
mbre,Personne,GFUN, Pnbr,Saillance)).

chercheP([phrase(X,gv(vid))],pronom(Val,Genre,Nombre,Personne,GFUN, Pnb
r),Discour):-
```

```
        Discour=[phrase(X,Y)|RestDiscour],
        Y=gv(V,gn(pronom(Val,Genre,Nombre,Personne,GFUN, Pnbr),Genre,No
mbre,Personne,GFUN, Pnbr,Saillance),Z).
/*-------------------------------------------------------------------------*/
chercheP([phrase(X,Y)|Avant],pronom(Val,Genre,Nombre,Personne,GFUN, Pn
br),Discour):-
        Discour=[phrase(X,Y)|RestDiscour],
        chercheP(Avant,pronom(Val,Genre,Nombre,Personne,GFUN, Pnbr),Rest
Discour).
/*-------------------------------------------------------------------------*/
chercheNP([],[]).

chercheNP([gn(X,Genre,Nombre,Personne,GFUN, NPnbr, Saillance)|Y],Discour
):-
        Discour=[phrase(gn(X,Genre,Nombre,Personne,GFUN, NPnbr, Saillance)
,gv(V))|Reste],
        chercheNP(Y,Reste).

chercheNP([gn(X1,Genre,Nombre,Personne,GFUN, NPnbr, Saillance)|[gn(X2,G
enre2,Nombre2,Personne2,GFUN2, NPnbr2, Saillance2)|Y]],Discour):-
        Discour=[phrase(gn(X1,Genre,Nombre,Personne,GFUN, NPnbr, Saillance
),gv(V,gn(X2,Genre2,Nombre2,Personne2,GFUN2, NPnbr2, Saillance2)))|Reste
],
        chercheNP(Y,Reste).

chercheNP([gn(X,Genre,Nombre,Personne,GFUN, NPnbr, Saillance)|Y],Discour
):-
        Discour=[phrase(gn(X,Genre,Nombre,Personne,GFUN, NPnbr, Saillance)
,gv(V,prep(Prep)))|Reste],
        chercheNP(Y,Reste).
```

```
chercheNP([gn(X1,Genre,Nombre,Personne,GFUN, NPnbr, Saillance)|[gn(X2,G
enre2,Nombre2,Personne2,GFUN2, NPnbr2, Saillance2)|Y]],Discour):-
        Discour=[phrase(gn(X1,Genre,Nombre,Personne,GFUN, NPnbr, Saillance
),gv(V,gn(X2,Genre2,Nombre2,Personne2,GFUN2, NPnbr2, Saillance2),Prep))|
Reste],
        chercheNP(Y,Reste).
/*-------------------------------------------------------------------------------*/
accord (En_accord, P, Liste_condidats):-
        P=pronom(Val_P,Genre,Nombre,Personne,Fonction2,Pnbr),
        not Genre=i,
        accord1(En_accord, P, Liste_condidats).

accord (En_accord, P, Liste_condidats):-
        P=pronom(Val_P,i,Nombre,Personne,Fonction2,Pnbr),
        accord2(En_accord, P, Liste_condidats).

accord1 ([],P,[]).

accord1 ([N|Suite_en_accord], P, Liste_condidats):-
        Liste_condidats = [N|Suite],
        P=pronom(Val_P,Genre,Nombre,Personne,Fonction2,Pnbr),
        N=gn(Val_Np,Genre,Nombre,Personne,Fonction,NPnbr,Saillance ),
        accord1 (Suite_en_accord, P, Suite).

accord1 (Suite_en_accord, P, Liste_condidats):-
        Liste_condidats = [N|Suite],
        P=pronom(Val_P,Genrep,Nombrep,Personnep,Fonction2,Pnbr),
        N=gn(Val_Np,Genrenp,Nombrenp,Personnenp,Fonction,NPnbr,Saillance
),
        not (Genrenp=Genrep,Nombrenp=Nombrep,Personnenp=Personnep),
        accord1 (Suite_en_accord, P, Suite).
```

```
accord2 ([],P,[]).

accord2 ([N|Suite_en_accord], P, Liste_condidats):-
        Liste_condidats = [N|Suite],
        P=pronom(Val_P,i,Nombre,Personne,Fonction2,Pnbr),
        N=gn(Val_Np,Genrenp,Nombre,Personne,Fonction,NPnbr,Saillance ),
        accord2 (Suite_en_accord, P, Suite).

accord2 (Suite_en_accord, P, Liste_condidats):-
        Liste_condidats = [N|Suite],
        P=pronom(Val_P,i,Nombrep,Personnep,Fonction2,Pnbr),
        N=gn(Val_Np,Genrenp,Nombrenp,Personnenp,Fonction,NPnbr,Saillance
),
        not (Nombrenp=Nombrep,Personnenp=Personnep),
        accord2 (Suite_en_accord, P, Suite).
/*-------------------------------------------------------------------------*/

best(Liste,N,P):-
          calcule([N|[]],Liste,P).

best(Liste,NPSP,P):-
          calcule(Liste_calcule,Liste,P),
          parallel_synt(NPP, Liste_calcule, P),
          not NPP=[],
          max(NPSP,NPP).

best(Liste,NPSP,P):-
          calcule(Liste_calcule,Liste,P),
          parallel_synt(NPP, Liste_calcule, P),
          NPP=[],
          max(NPSP,Liste_calcule).
/*-------------------------------------------------------------------------*/
```

```
parallel_synt ([], [], P).

parallel_synt ([NP|NPSUITE], LESNP, P):-
        LESNP = [NP|Suite],
        NP=gn(Val_Np,Genre,Nombre,Personne,Fonction,NPnbr,Saillance),
        P=pronom(Val_P,Genre2,Nombre,Personne,Fonction,Pnbr),
        parallel_synt (NPSUITE,Suite, P).

parallel_synt (NPSUITE, LESNPS, P):-
       LESNPS = [NP|Suite],
       NP=gn(Val_Np,Genre,Nombre,Personne,Fonction1,NPnbr,Saillance),
       P=pronom(Val_P,Genre2,Nombre,Personne,Fonction2,Pnbr),
       not Fonction1= Fonction2,
       parallel_synt (NPSUITE, Suite, P).

/*-------------------------------------------------------------------------------*/
max([NP|[]], [NP|[]]).

max([],[]).

max([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)|[]], L):-
   L=[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)|L2],
   max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,Saillance2)|
NPS],L2),
   Saillance > Saillance2.

max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,Saillance2)|NP
S], L):-
   L=[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)|L2],
   max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,Saillance2)|
NPS],L2),
```

```
   Saillance < Saillance2.

max([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)|
   [gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,Saillance2)|NPS]]
, L):-
   L=[gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance)|L2],
   max([gn(Val_Np2,Genre2,Nombre2,Personne2,GFUN2,NPnbr2,Saillance2)|
NPS],L2),
   Saillance = Saillance2.

/*-------------------------------------------------------------------------*/

calcule (Npscalcule,Npsnoncalcule,pronom(Val_P,Genre,Nombre,Personne,Fon
ction,Pnbr)) :-
   calculeGFUN(NPs_Gfun_calc,Npsnoncalcule) ,
   recent(Npscalcule,NPs_Gfun_calc,Pnbr).
/*-------------------------------------------------------------------------*/
calculeGFUN([],[]).
/*----------------SUBJ-S: 80 iff GFUN = subject ------------------------------*/

calculeGFUN ([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance2)|R
estcalcule],
      [gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1)| Rest]) :-
   GFUN= subject,
   Saillance2= Saillance1+80,
   calculeGFUN (Restcalcule, Rest).

/*----------------------POSS-S: 65 iff GFUN = possessive ---------------------*/

calculeGFUN ([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance2) |
Restcalcule],
          [gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1)| Rest])
```

```
 :-
    GFUN= possessive,
    Saillance2= Saillance1+65,
    calculeGFUN (Restcalcule, Rest).

/*-------------ACC-S: 50 iff GFUN = direct object-----------*/

calculeGFUN ([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance2) |
Restcalcule],
        [gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1)| Rest]) :-
    GFUN= directobject,
    Saillance2= Saillance1+50,
    calculeGFUN (Restcalcule, Rest).

/*-----------------DAT-S: 40 iff GFUN = indirect object-------------------------*/

calculeGFUN ([gn(Val_Np,Genre,Nombre,Personne,GFUN, NPnbr,Saillance2) |
Restcalcule],
        [gn(Val_Np,Genre,Nombre,Personne,GFUN, NPnbr,Saillance1)| Rest]) :-
    GFUN= indirectobject,
    Saillance2= Saillance1+40,
    calculeGFUN (Restcalcule, Rest).

/*--------------------SENT-S: 100 iff in the current sentence---------------------*/

recent([],[],Pnbr).
recent([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance2) | Restaca
lculer],
        [gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1) | Rest], Pn
br) :-
    Pnbr < NPnbr+10,
    recent (Restacalculer, Rest, Pnbr),
```

```
    Saillance2= Saillance1+(10+NPnbr- Pnbr)*10.

recent ([gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1) | Restac
alculer],
        [gn(Val_Np,Genre,Nombre,Personne,GFUN,NPnbr,Saillance1) | Rest], Pn
br) :-
    Pnbr >= NPnbr+10,
    recent (Restacalculer, Rest, Pnbr).
/*----------------------------------Fin-------------------------------------------*/
```

Résumé

La résolution des liens anaphorique est importante pour la « compréhension» d'un texte en langue naturelle par un ordinateur. Dans ce travail, nous nous intéressons aux anaphores pronominales dont l'antécédent est un groupe nominal.

La résolution consiste à parcourir la représentation syntaxique en recherchant les pronoms et les groupe nominaux qui les précèdent. Ensuite, on éliminera les candidats selon des critères morphosyntaxiques, s'il nous reste plus qu'un candidat en sélection un selon des préférences. Et pour tester notre algorithme nous l'avons implémenté en Prolog.

Mots-clés : anaphores pronominales, résolution d'anaphores pronominales, chaîne de coréférence, filtre syntaxique, calcul de saillance.

Abstract

The anaphora resolution is important for "understanding" the natural language text by computer. In this work, we work on pronominal anaphora in which the antecedent is a noun phrase.

In the resolution we browse the syntactic representation searching for pronouns and noun phrases that precede them. Then we eliminate candidates according to morphosyntactic filter, if we have more than one candidat. we chose the most likely according to preferences. To test our algorithm we have implemented it in Prolog

Keywords: pronominal anaphora, pronominal anaphora resolution, coreference
chain, syntactic filter, calculation of salience.