Le TALN et Le RTE

1) Introduction

Dans ce chapitre, nous commencerons par clarifier quelques concepts linguistiques, en étudiant les différents niveaux de représentation et de traitement des énoncés linguistiques. La section suivante est consacrée à l'étude de l'inférence textuelle où nous présentons les différentes applications du RTE et les principaux niveaux d'inférences textuelles nous détaillons les étapes de développement du challenge Pascale RTE qui a été mis en œuvre pour évaluer les avances des groupes de recherches dans ce domaine.
Nous terminons ce chapitre par la présentation de quelques méthodes d'inférences utilisées par des groupes de recherches évaluées dans le challenge Pascal RTE.

2) Brève historique du traitement automatique du langage naturel

Historiquement, les premiers travaux importants dans le domaine du TALN ont porté sur la traduction automatique, avec, dès 1954, la mise au point du premier traducteur automatique (très rudimentaire). Quelques phrases russes, sélectionnées à l'avance, furent traduites automatiquement en anglais.

Depuis 1954, de lourds financements ont été investis et de nombreuses recherches ont été lancées. Les principaux travaux présentés concernent alors la fabrication et la manipulation de dictionnaires électroniques, car les techniques de traduction consistent essentiellement à traduire mot à mot, avec ensuite un éventuel réarrangement de l'ordre des mots.

Cette conception simpliste de la traduction a conduit à l'exemple célèbre suivant : la phrase *The spirit is willing but the flesh is weak* (l'esprit est fort mais la chair est faible) fut traduite en russe puis retraduite en anglais.
Cela donna quelque chose comme : *The vodka is strong but the meat is rotten* (la vodka est forte mais la viande est pourrie) !

Ce qui ressort de cet exemple, c'est que de nombreuses connaissances contextuelles (i.e. portant sur la situation décrite) et encyclopédiques (i.e. portant sur le monde en général) sont nécessaires pour trouver la traduction correcte d'un mot (par exemple ici spirit, qui, suivant les contextes peut se traduire comme esprit ou comme alcool).

Posant comme conjecture que tout aspect de l'intelligence humaine peut être décrit de façon suffisamment précise pour qu'une machine le simule, les figures les plus marquantes de l'époque (John Mc Carthy, Marvin Minsky, Allan Newell, Herbert Simon) y discutent des

possibilités de créer des programmes d'ordinateurs qui se comportent intelligemment, et en particulier qui soient capables d'utiliser le langage.

Aujourd'hui, le champ du traitement du langage naturel est un champ de recherche très actif. De nombreuses applications industrielles (traduction automatique, recherche documentaire, interfaces en langage naturel), qui commencent à atteindre le grand public, sont là pour témoigner de l'importance des avancées accomplies mais également des progrès qu'il reste encore à accomplir.

3) Les niveaux de traitement

Nous introduisons dans cette section les différents niveaux de traitements nécessaires pour parvenir à une compréhension complète d'un énoncé en langage naturel. Ces niveaux correspondent à des modules qu'il faudrait développer et faire coopérer dans le cadre d'une application complète de traitement de la langue.

Nous considérons à titre d'exemple l'énoncé suivant :

(1) Le président des antialcooliques mangeait une pomme avec un couteau,

Nous envisageons les traitements successifs qu'il convient d'appliquer à cet énoncé pour parvenir automatiquement à sa compréhension la plus complète. Il nous faudra successivement :
– identifier les composants lexicaux, et leurs propriétés : c'est l'étape de traitement **lexical** ;
– identifier des constituants (groupe) de plus haut niveau, et les relations (de dominance) qu'ils entretiennent entre eux : c'est l'étape de traitement **syntaxique** ;
– construire une représentation du sens de cet énoncé, en associant à chaque concept évoqué un objet ou une action dans un monde de référence (réel ou imaginaire) : c'est l'étape de traitement **sémantique**.
– identifier enfin la fonction de l'énoncé dans le contexte particulier de la situation dans lequel il a été produit : c'est l'étape de traitement **pragmatique.**

3.1) Le niveau lexical

Le but de cette étape de traitement est de passer des formes atomiques (tokens) identifiées par le segmenteur de mots (Nugues, 2006), c'est-à-dire de reconnaître dans chaque chaîne de caractères une (ou plusieurs) unité(s) linguistique(s), dotée(s) de caractéristiques propres (son sens, sa prononciation, ses propriétés syntaxiques, etc).

Selon l'exemple *(1)*, l'étape d'identification lexicale devrait conduire à un résultat voisin de celui donné ci-dessous, dans lequel on peut constater en particulier l'ambiguïté d'une forme telle que président: cette chaîne correspond à deux formes du verbe présider (indicatif et subjonctif), ainsi à une forme nominale, et sa prononciation diffère selon qu'elle représente un nom ou un verbe.

On conçoit aisément que pour les mots les plus fréquents, comme « le », la solution la plus simple est de rechercher la forme dans (un lexique)[1] précompilé. Dans les faits, c'est effectivement ce qui se passe, y compris pour des formes plus rares, dans la mesure où l'utilisation des formalismes de représentations compacts permettant un accès optimisé (par exemple sous la forme d'automates d'états finis), et l'augmentation de la taille des mémoires rend possible la manipulation de vastes lexiques (de l'ordre de centaines de milliers de formes).
Pour autant, cette solution ne résout pas tous les problèmes. Le langage est création, et de nouvelles formes surgissent tous les jours, que ce soit par emprunt à d'autres langues (il n'y a qu'a écouté parler les enseignants des autres modules de la dominante informatique !), ou, plus fréquemment, par l'application de procédés réguliers de créations de mots, qui nous permettent de composer pratiquement à volonté de nouvelles formes immédiatement compréhensibles par tous les locuteurs de notre langue : si j'aime lire Proust, ne peut-on pas dire que je m'emproustise, que de proustien je deviens proustiste, voire proustophile, puis que, lassé, je me désemproustise... Ce phénomène n'a rien de marginal, puisqu'il est admis que, même si l'on dispose d'un lexique complet du français, environ 5 à 10 % des mots d'un article de journal pris au hasard ne figureront pas dans ce lexique. La solution purement lexicale atteint là ses limites, et il faut donc mettre en œuvre d'autres approches, de manière à traiter aussi les formes hors-lexiques.

3.2) Le niveau syntaxique

La syntaxe est l'étude des contraintes portant sur les successions licites de formes qui doivent être prises en compte lorsque l'on cherche à décrire les séquences constituant des phrases grammaticalement correctes: toutes les suites de mots ne forment pas des phrases acceptables (Ligauzat, 1994).
La description des contraintes caractéristiques d'une langue donnée se fait par le biais d'une grammaire.

Les modèles et les formalismes grammaticaux proposés dans le cadre du traitement automatique du langage sont particulièrement nombreux et variés.
Le niveau syntaxique est donc le niveau conceptuel concerné par le calcul de la validité de certaines séquences de mots, les séquences grammaticales ou bien-formées. On conçoit bien l'importance d'un tel traitement dans une application de génération, pour laquelle il est essentiel que la machine engendre des énoncés corrects. Dans une application de compréhension, la machine analyse des textes qui lui sont fournis, et dont on peut supposer qu'ils sont grammaticaux. Pourquoi donc, dans ce cas, mettre en œuvre des connaissances syntaxiques ?

Une première motivation provient du fait que les textes ne sont pas toujours grammaticaux, par exemple à cause des fautes d'orthographes. Une analyse syntaxique peut donc permettre de choisir entre plusieurs corrections à apporter à une phrase incorrecte, mais également se révéler bien utile pour améliorer les sorties d'un système de reconnaissance optique de caractère ou d'encore un système de reconnaissance de la parole.

[1] En linguistique, le lexique d'une langue constitue l'ensemble de ses lemmes ou, d'une manière plus courante mais moins précise, « l'ensemble de ses mots ». Toujours dans les usages courants, on utilise, plus facilement le terme *vocabulaire*.

Une seconde raison est que l'entrée du module syntaxique est une série de formes étiquetées morpho syntaxiquement, une forme pouvant avoir plusieurs étiquettes différentes. Une première fonction du module syntaxique consiste donc à désambiguïser la suite d'étiquettes, en éliminant les séquences qui correspondent à des énoncés grammaticalement invalides.

3.3) Le niveau sémantique

Intuitivement, la sémantique se préoccupe du sens des énoncés (yvon, 2007). Une phrase comme *Le jardin de la porte mange le ciel*, bien que grammaticalement parfaitement correcte, n'a pas de sens dans la plupart des contextes. Mais qu'est ce que le sens ? Pour une expression comme *la bouteille de droite* dans la phrase :

Sers-toi du vin. Non, pas celui-là, prends la bouteille de droite.

Le sens correspond à l'objet (au concept) désigné. Dans cet exemple, le sens dépend étroitement du contexte : il faut une représentation de la scène pour savoir de quelle bouteille, et donc de quel vin, il s'agit.

Pour une expression prédicative, comme *Il commande un Margaux 1982*, le sens peut être représenté par un prédicat logique comme <demander(paul,chateau_margaux_82)>.
L'identification d'un tel prédicat dépend encore une fois du contexte. Le verbe commander aurait en effet renvoyé à un autre prédicat s'il s'agissait de commander un navire.

3.4) Le niveau pragmatique

Le niveau pragmatique est parfaitement dissociable du niveau sémantique. Alors que la sémantique se préoccupe du sens des énoncés, la pragmatique porte sur les attitudes (vérité, désirabilité, probabilité) que les locuteurs adoptent vis à vis des énoncés et sur les opérations logiques que ces attitudes déclenchent (yvon, 2007).

Historiquement, certains linguistes ont appelé pragmatique tout traitement du langage faisant intervenir le contexte d'énonciation. Ce critère présente fort peu d'intérêt, dans la mesure où les processus sémantiques sont les mêmes, que le contexte intervienne ou non. En revanche, il existe une distinction très importante, basée sur la notion d'inférence logique. Considérons l'exemple suivant :

(a) Pierre : viendras-tu au bal ce soir ?

(b) Marie : j'ai entendu que Paul y sera !

La seconde phrase sera interprétée comme une réponse négative si l'on sait que *Marie* n'aime pas *Paul*.

Cette interprétation n'est pas de nature sémantique. À partir de la compréhension du sens de l'intervention de *Marie*, *Pierre* réalise une inférence logique en utilisant une connaissance contextuelle, l'inimitié entre Paul et Marie. Pierre conclut que Marie ne veut pas aller au bal, autrement dit il reconstruit l'attitude de Marie par rapport à son propre énoncé. Cette opération n'est pas une construction conceptuelle, c'est une opération logique. Elle appartient donc à la pragmatique.

Les techniques correspondant à ce niveau de traitement sont encore très mal maîtrisées. Le niveau pragmatique, même si les techniques qui lui correspondent ne sont pas encore stabilisées, apparaît moins difficile à aborder que le niveau sémantique. Il semble en effet qu'il repose sur un ensemble de principes fixes, comme le principe de pertinence, qu'il s'agit de modéliser correctement. La détermination de l'intention argumentative de l'auteur ou du locuteur est essentielle dans bon nombre d'applications, notamment la gestion de dialogue, le résumé de texte, la traduction automatique, les systèmes d'aide contextuelle ou d'enseignement, etc. On attend donc des progrès significatifs à ce niveau dans les années qui viennent.

4) Les difficultés du TALN : ambiguïté

Le langage naturel est ambigu, et cette ambiguïté se manifeste par la multitude d'interprétations possibles pour chacune des entités linguistiques pertinentes pour un niveau de traitement, comme en témoignent les exemples suivants :

4.1) Ambiguïté des graphèmes (lettres)

Cette ambigüité existe dans le processus d'encodage orthographique en comparant la prononciation du i dans lit, poire et maison.

4.2) Ambiguïté dans les propriétés grammaticales et sémantiques

Ainsi *mange* est ambigu à la fois morpho-syntaxiquement, puisqu'il correspond aux formes indicatives et subjonctives du verbe manger), mais aussi sémantiquement. En effet, cette forme peut aussi bien référer (dans un style familier) à un ensemble d'actions conventionnelles (comme de s'assoir à une table, mettre une serviette, utiliser divers ustensiles, ceci éventuellement en maintenant une interaction avec un autre humain) avec pour vision finale d'ingérer de la nourriture (auquel il ne requière pas de complément d'objet direct); et à l'action consistant à effectivement ingérer un type particulier de nourriture (auquel cas il requiert un complément d'objet direct), etc. Comparez en effet :

(a) Demain, Paul mange avec ma sœur.
(b) Paul mange son pain au chocolat.

Ainsi que les déductions que l'on peut faire à partir de ces deux énoncés : de (a), on peut raisonnablement conclure que Paul sera assis à une table, disposera de couverts,... ; tout ceci n'est pas nécessairement vrai dans le cas de l'énoncé (b).

4.3) Ambiguïté de la fonction grammaticale des groupes de mots

L'ambigüité est illustrée par la phrase :
il poursuit la jeune fille à vélo.
Dans cet exemple *à vélo* est soit un complément de manière de poursuivre (et c'est *il* qui pédale), soit un complément de nom de fille (et c'est *elle* qui mouline) ;

4.4) Ambiguïté de la portée des quantificateurs, des conjonctions et des prépositions

Ainsi, dans *Tous mes amis ont pris un verre,* nous pouvons supposer que chacun avait un verre différent, mais dans *Tous les témoins ont entendu un cri,* il est probable que c'était le même cri pour tous les témoins. De même, lorsque l'on évoque les chiens et les chats de Paul, l'interprétation la plus naturelle consiste à comprendre de Paul comme le complément de nom du groupe les chats et les chiens ; cette lecture est beaucoup moins naturelle dans les chiens de race et les chats de Paul ;

4.5) Ambiguïté sur l'interprétation à donner en contexte à un énoncé

Nous comparons ainsi la « signification » de non, dans les deux échanges suivants :

(a) Si je vais en cours demain ? Non (négation)
(b) Tu vas en cours demain ! Non ! (j'y crois pas).

En effet, l'ambiguïté est un problème majeur du TALN. Pour y pallier les chercheurs ont crée un domaine qui a pour but de centraliser ce problème et de proposer des méthodes de traitement du langage au niveau lexical, syntaxique et sémantique indépendamment d'une application donnée. Dans ce qui suit nous allons explorer ce domaine ainsi que ces différentes applications.

5) La reconnaissance de l'inférence textuelle (RTE)

5.1) Introduction

Le RTE est un domaine de recherche assez récent en traitement du langage (2005) qui a pour but de fédérer les recherches en TALN afin de proposer des méthodes de traitement du langage au niveau lexical, syntaxique et sémantique indépendamment d'une application donnée (résumé automatique, système de question réponse ou encore la recherche d'information).
Le RTE vise à déterminer automatiquement si un segment de texte (H) est déduit d'un autre segment de texte (T) (Dagan et al, 05).

Exemple :

T : « Amine a 40 degrés de ***fièvre****, sa mère l'a pris immédiatement à l'****hôpital*** *».*

H : « Amine est **malade** *».*

Dans l'exemple ci dessus, comprendre que le segment H est déduit du segment T, est une déduction simple pour l'être humain, mais pour la machine c'est tout autre. Pour cela, les chercheurs ont proposé plusieurs approches pour résoudre le problème.

Dans l'exemple, pour dire que H est inféré de T le système doit lier le fait d'être malade (texte H) avec le mot hôpital et fièvre (texte T) pour déduire qu'il y a inférence.

Dans cette section, nous présentons les différentes applications du RTE, puis nous détaillons les étapes de développement du challenge Pascale RTE qui a été mis en œuvre pour évaluer les avances des groupes de recherches dans ce domaine.

Nous développons dans la section 2, les principaux niveaux d'inférences textuelles et nous terminons ce chapitre par la présentation de quelques méthodes d'inférences utilisées par des groupes de recherches évaluées dans le challenge pascal RTE.

5.2) Les applications du RTE

L'inférence entre des segments de textes est au cœur de plusieurs applications du traitement automatique du langage naturel (TALN). Nous décrivons dans ce qui suit comment le RTE contribue dans ces différents domaines :

5.2.1) La recherche d'information

La recherche d'information est la science qui consiste à rechercher l'information dans des documents, des bases de données, qu'elles soient relationnelles ou mises en réseau par des liens hypertextes (Joachims, 2003).

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie qui ont toujours eu le souci d'établir des représentations des documents dans le but d'en récupérer des informations, à travers la construction d'index. L'informatique a permis le développement d'outils pour traiter l'information et à établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher l'information.
Les approches qui étaient utilisées auparavant se basaient sur la recherche de mots clés dans les textes. Le problème dans ces systèmes c'est qu'ils ne prennent en compte ni les relations entre les mots clés ni leurs sens.

Exemple 1 :

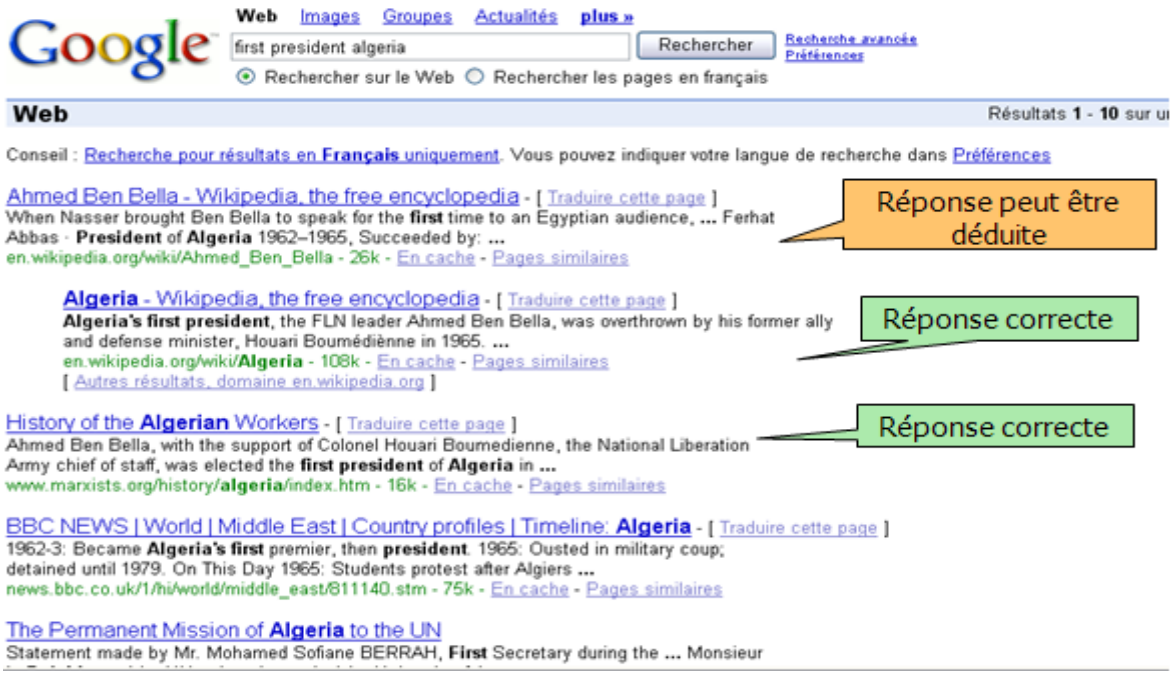

Figure 1.1 : Exemple de moteur de recherche a base de mot clé

Dans cet exemple (Figure 1.1) nous remarquons qu'un moteur de recherche fonctionnant à base de mot clé comme Google fait bien ce type de recherche et répond bien à la question simple comme « the first president Algerie » puisque la simple recherche des mots clés dans les différents documents permet de donner une bonne réponse à l'utilisateur.

Exemple 2 :

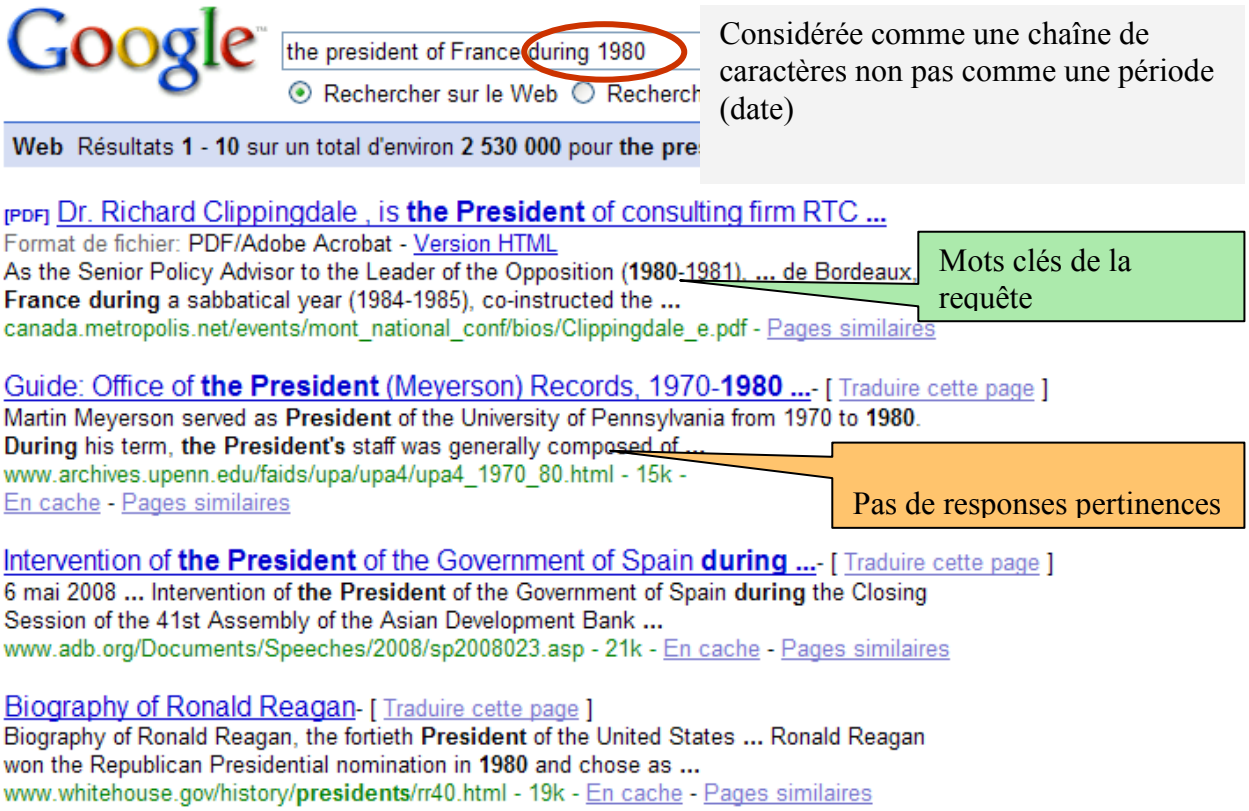

Figure 1.2 : Exemple où le moteur de recherche à base de mot clé ne marche pas

Dans cet exemple (Figure 1.2) nous remarquons que l'utilisation des mots clés seuls peut nous mener à un document qui n'a aucune relation avec notre requête et qui montre que l'inférence sémantique est indispensable à la recherche d'information.

5.2.2) L'extraction d'information

L'extraction d'information consiste à identifier l'information bien précise d'un texte en langue naturelle et à la représenter sous forme structurée. Par exemple, à partir d'un rapport sur un accident d'automobile, un système d'extraction d'information sera capable d'identifier la date et le lieu de l'accident, le type d'incident, ainsi que les victimes. Ces informations pourront ensuite être stockées dans une base de données pour y effectuer des recherches ultérieures ou être utilisées comme base à la génération automatique de résumés (Kosseim., 2005).

L'extraction d'information s'avère très pratique dans l'industrie où des opérations d'extractions y sont quotidiennement effectuées à la main. Nous pensons, par exemple, au traitement de rapports de filature d'une agence de surveillance, à la gestion de dépêches d'une agence de presse, à la manipulation de rapports d'incidents d'une compagnie d'assurances, etc.
Un système d'extraction d'information permet de traiter automatiquement et plus rapidement de grandes quantités de documents.

Dans ce cas de figure le RTE donne son apport dans la détection de l'information.

5.2.3) Le système question- réponse

Les systèmes Questions/Réponses sont capables de répondre à des questions écrites en langage naturel en cherchant la réponse dans un corpus de textes. Ils sont classiquement constitués d'un ensemble de modules réalisant respectivement : une analyse de la question, une recherche de portions de documents pertinents et une extraction de la réponse à l'aide de motifs d'extractions, ou patterns en anglais (Nyberg et al, 2002).
Le système doit identifier le segment de texte qui contient la réponse. L'inférence entre le texte T et le segment H peut aider à détecter le segment qui contient la réponse.

Exemple :
H : « who is Ariel Sharon ? ».
T : « Israel's Prime Minister, Ariel Sharon, visited Prague ».
Le système effectue d'abord une transformation à l'affirmatif de la question « Ariel Sharon is Isreal's Prime Minister » puis une comparaison entre le segment de texte T et le segment H.
Si H est inféré de T comme dans l'exemple alors T est accepté comme un segment contenant la réponse à la question H.

5.2.4) La traduction automatique

La traduction automatique désigne, au sens strict, le fait de traduire entièrement un texte grâce à un ou plusieurs programmes informatiques, sans qu'un traducteur humain n'ait à intervenir (Laurian et Marie, 1996). La traduction automatique est encore très imparfaite, et la génération de traduction d'une qualité comparable à celle de traducteurs humains relève encore de l'utopie.

Pour évaluer les performances de la machine, le RTE permet de comparer la traduction faite par la machine avec celle faite par l'humain.

5.2.5) Le résumé automatique

Le résumé automatique se propose de faire une extraction de l'information jugée importante d'un texte d'entré pour construire, à partir de cette information, un nouveau texte de sortie, condensé. Ce nouveau texte permet d'éviter la lecture en entier du document source.

Le RTE est utilisé pour trouver les redondances d'informations.
Si un segment de texte infère un autre, un des deux va être supprimé.
En particulier c'est intéressant dans les applications qui font le résumé de plusieurs documents. S'il y a plusieurs documents qui relatent le même fait, un seul doit être pris.

5.2.6) L'acquisition des Paraphrases (AP)

Une paraphrase, c'est le fait de dire avec d'autres mots, d'autres termes ce qui est dit dans un texte, un paragraphe.

Dans ce cas de figure le RTE est utilisé pour détecter l'inférence entre le texte paraphrasé et le texte d'origine. Comme dans l'exemple suivant où les deux phrases ont le même sens avec juste une autre disposition des mots dans la phrase.
Exemple :
T : « Ce médicament est commercialisé au Canada seulement ».
H **:** « La commercialisation de ce médicament s'est effectuée au Canada seulement ».

5.3) Le challenge "PASCAL Recognizing of Textual Entailment"

Le Pascal recognition of Textual Entailment est un concours qui a débuté en 2005. Il se déroule chaque année et son objectif, est de fournir à la communauté du TAL un nouveau point de repère pour vérifier les progrès dans la reconnaissance l'inférence textuelle, et de comparer les réalisations des différents groupes de recherches travaillant dans ce domaine (http://www.pascal-network.org/Challenges/RTE/).

Suite au succès du premier RTE un nouveau RTE a été organisé, avec 23 groupes venus du monde entier (par rapport à 17 pour le premier défi) qui ont présenté les résultats de leurs systèmes. Les représentants des groupes participants ont présenté leurs travaux au **PASCAL Challenges atelier** en avril 2006 à Venise, Italie.

L'événement a été un succès et le nombre de participants et leurs contributions à la discussion ont démontré que le Textual Entailment est un domaine en expansion rapide. Déjà, les ateliers ont donné naissance à un nombre impressionnant de publications dans les grandes conférences, en plus des travaux en cours.
Les démarches entreprises pour réaliser le concours sont :

- Préparation du corpus.
- Etablissement des mesures d'évaluations.

Dans ce qui suit les démarches citées sont détaillées.

5.3.1) La préparation du corpus

La première étape à entreprendre consiste à créer le corpus de texte-hypothèse (T-H) pair de petit segment de texte, qui correspond à des informations collectées à travers le web dans des domaines différents.
Les exemples ont été collectés manuellement pour l'inférence par des annotateurs humains.
Les exemples ont été divisés en deux types de corpus (**Corpus de développement** et **Corpus de test)**.

Le corpus de développement est utilisé au début de challenge pour donner aux utilisateurs la possibilité de tester leurs systèmes et de faire des petites mises au point pour se préparer au test.

Le corpus de test est utilisé pour l'évaluation finale.

1. Pour le RTE 1 Le corpus était composé de 567 paires de (H-T) pour le développement et 800 pairs pour le test.

Le choix d'un large corpus est justifié par la nécessité d'avoir des résultats statistiques significatifs.

Le corpus est collecté en respectant les différentes applications du traitement de langage naturel (QR, RI, IE., PP…) et la collecte des exemples est faite par niveau d'inférence :
L'analyse lexique, syntaxique, logique et connaissance du monde, et les différents niveaux de difficultés.

```
<pair id="754" value="TRUE" task="CD">

    <t>

Mexico City has a very bad pollution problem because the mountains
    around the city act as walls and block in dust and smog.

</t>

    <h> Poor air circulation out of the mountain-walled Mexico
    City aggravates pollution.</h>

</pair>

Id : représente le numéro de la pair.

Value : représente la décision de l'annotateur (vrai ou faux).

Task : représente le type de l'application ou l'inférence existe.
```

Figure 1.3 : Exemple du corpus annoté

Le corpus doit inclure 50% d'un exemple de T-H correspondant à de vraies inférences et 50% de fausses inférences. Pour cela, chaque exemple (T-H) est jugé vrai ou faux par l'annotateur qui crée l'exemple.
Puis l'exemple est évalué par un second juge qui évalue les paires de textes et d'hypothèses, sans avoir pris conscience de leurs contextes.

Les annotateurs étaient d'accord avec le jugement dans 80% des exemples, ce qui correspond à 0.6 Kappa[2], les 20% du corpus où il n'y a pas eu d'accord ont été supprimés). Le reste du corpus est considéré comme un «gold standard» ou « BASELINE » pour l'évaluation.

Le but de cette manœuvre est de créer un corpus où il n'y aura pas de jugements controverses. Pour effectuer leurs jugements et annoter le corpus les annotateurs suivent des directives. Dans ce qui suit, nous allons citer les différentes directives qui étaient prises en considération.

5.3.2) Les directives de jugements

- L'inférence est une relation à un seul sens.
- L'hypothèse doit être inférée d'un texte, mais le texte ne doit pas forcement être inféré de l'hypothèse.

- L'hypothèse doit être inférée entièrement du texte. L'inférence est fausse s'il reste une partie de l'hypothèse qui ne peut être inférée par le texte.

[2] Kappa (J.Cohen, 1960) :c'est une mesure statistique pour calculer a quel point deux personnes (ou groupes de personnes) A et B sont d'accord pour classer N éléments dans K catégories mutuellement exclusives.

- les cas où l'inférence est probable doit être jugé comme vrai.

- il est autorisé d'utiliser les connaissances du monde comme dans l'exemple *le chiffre d'affaire de Google est de 50 millions de dollars*. On doit savoir que Google est une entreprise donc on peut lui attribuer la possibilité d'avoir un chiffre d'affaire.

5.3.3) Les mesures d'évaluation

Le système d'annotation du corpus adopté dans les deux challenges précédant est binaire, c'est-à-dire que le système donne deux résultats possibles soit l'inférence entre les deux textes est vrai ou fausse}.
Le résultat est comparé au '**GOLD standard'**, et le pourcentage donnant le nombre de fois où il y a similitude entre le système et le '**gold standard'** donne **'l'accuracy'** du système.
L'accuracy est une mesure standard dans les systèmes de traitement du langage naturel. Elle est fréquemment utilisée pour évaluer les performances des applications, (Beyer et al. 2005). Elle est calculée comme ceci.
Accuracy = X / Y.
Où :
X : représente le nombre de fois où les résultats du système sont similaires au gold standard.
Y : représente le nombre de paires contenu dans le corpus de test.
Par exemple Le nombre de résultats similaires est de 500 paires et le corpus est de 800 paires, l'accuracy est de 500/800 qui est égale à 62,5%.

5.4) L'analyse des principales méthodes utilisées

Dans ce qui suit, nous allons présenter les différentes étapes de traitements effectuées pour détecter l'inférence textuelle.

5.4.1) Les prétraitements

Quelque soit la technique adoptée pour effectuer l'inférence textuelle, il est nécessaire de pré traiter les données brutes avant d'appliquer les techniques d'inférences.
Dans le RTE trois niveaux de prétraitements ont été utilisés:

- Niveau lexical pour éviter les problèmes liés à la morphologie de mots.
- Niveau syntaxique pour pouvoir donner une structure préalable au texte.
- Niveau sémantique pour analyser les sens des mots.

Ci-dessous nous allons présenter les différents niveaux de prétraitements existants et utiliser pour l'inférence textuelle.

5.4.1.1) Le Niveau lexical

L'objectif du prétraitement au niveau du "mot" est de réduire les variations dues à la morphologie et d'éviter que des petites erreurs initiales se propagent dans toutes les étapes du traitement. Pour cela, différentes transformations ont été introduites :

A) La tokenisation

L'objectif de la tokenisation est de trouver les unités de base du "sens " dans les textes. Pour cela, les systèmes doivent résoudre différents problèmes comme la gestion des blancs, de la ponctuation, des retours lignes et des fins de paragraphes.

B) La lemmatisation

La lemmatisation d'une forme d'un mot consiste à en prendre sa forme canonique. Celle-ci est définie comme ceci :
Quand c'est un verbe on doit le mètre à l'infinitif :

Exemple :
Parti (verbe) -> partir

Pour les autres mots, ils doivent être mis au masculin singulier.

Exemple :
Parti (nom) -> parti

Pour effectuer l'analyse lexicale, différents outil qui ont été mis en point. Le TreeTagger est un des outils le plus utilisés pour la langue anglaise.

Le TreeTagger effectue une tokinisation, une lemmatisation et un étiquetage comme le montre l'exemple suivant :

Exemple d'entrée dans le **TreeTagger** : « `Le TreeTagger est facile à utiliser` ».

La figure suivante reprend la sortie du logiciel.

Tokenisation	**Étiquetage**	**Lemmatisation**
Le	DT	La
TreeTagger	NP	TreeTagger
Est	VBZ	Être
Facile	JJ	Facile
À	D'	À
Utiliser	VB	Utiliser

Figure 1.4 : Sortie du TreeTagger

5.4.1.2) Le niveau syntaxique

L'objectif de cette étape est de décrire les structures de phrases possibles et d'analyser les phrases en structures.
La structure révélée par l'analyse donne alors précisément la façon dont les règles syntaxiques sont combinées dans le texte. Cette structure est souvent une hiérarchie de syntagmes, représentée par un arbre syntaxique dont les nœuds peuvent être décorés (dotés d'informations complémentaires).

Nous illustrons cette analyse avec la sortie d'un des outils utilisés dans l'annotation syntaxique (SYNTEX)[3].

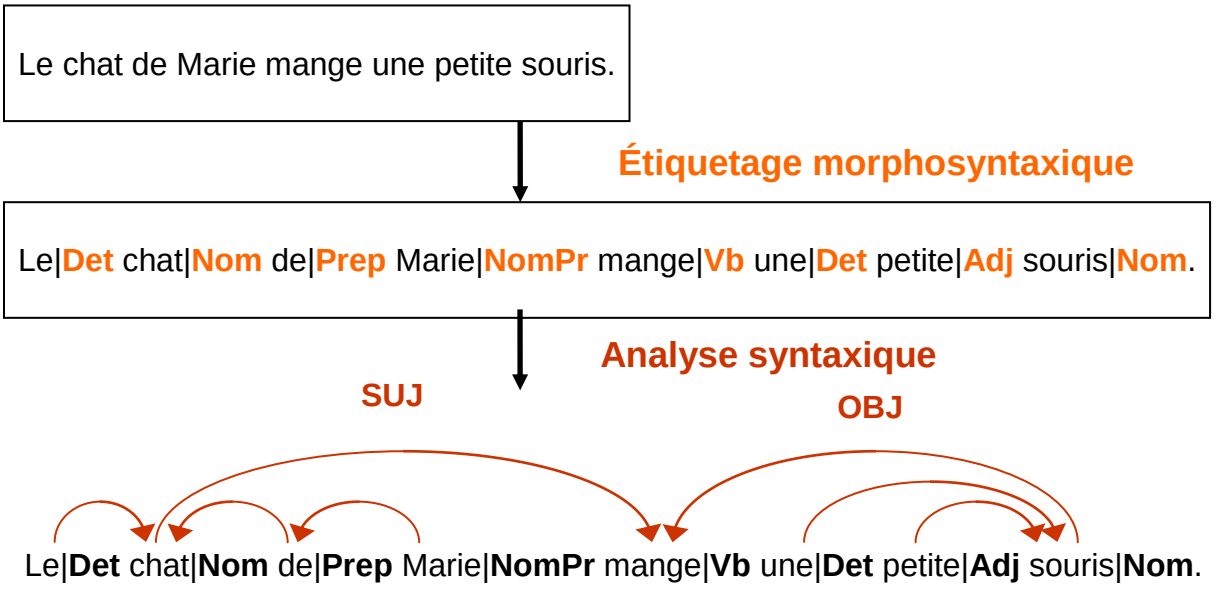

Figure 1.5 : Exemple d'annotation syntaxique

Nous remarquons dans l'exemple ci-dessus que l'analyse morphosyntaxique permet d'étiqueter les mots et l'analyse syntaxique permet de les relier entre eux.

5.4.1.3) Le niveau sémantique

Pour simplifier, nous pouvons dire que l'analyse sémantique s'appuie, entre autres, sur la compréhension du sens des mots des textes, contrairement aux analyses lexicales ou grammaticales, qui analysent les mots à partir du lexique ou de la grammaire. Dans le cadre de l'analyse sémantique, il est donc fondamental d'analyser le sens des mots pour comprendre ce qu'on dit. Pour cela plusieurs approches ont été adoptées pour annoter les relations entre les mots pour mieux cerner leur sens. Une de ces approches est la structure prédicat argument qui est expliquée ci-dessous.

La structure que nous appelons prédicative est un graphe de relation prédicat-argument, où les prédicats représentent l'action.
Une relation prédicative correspond à une relation de dépendance syntaxique. Le prédicat peut avoir plusieurs types d'arguments (sujet, complément d'objet direct et complément d'objet indirect).

[3] La fonction de cet analyseur est d'identifier des relations de dépendances entre mots et d'extraire d'un corpus des syntagmes (verbaux, nominaux, adjectivaux) (Bourigault, 2000).

Exemple :

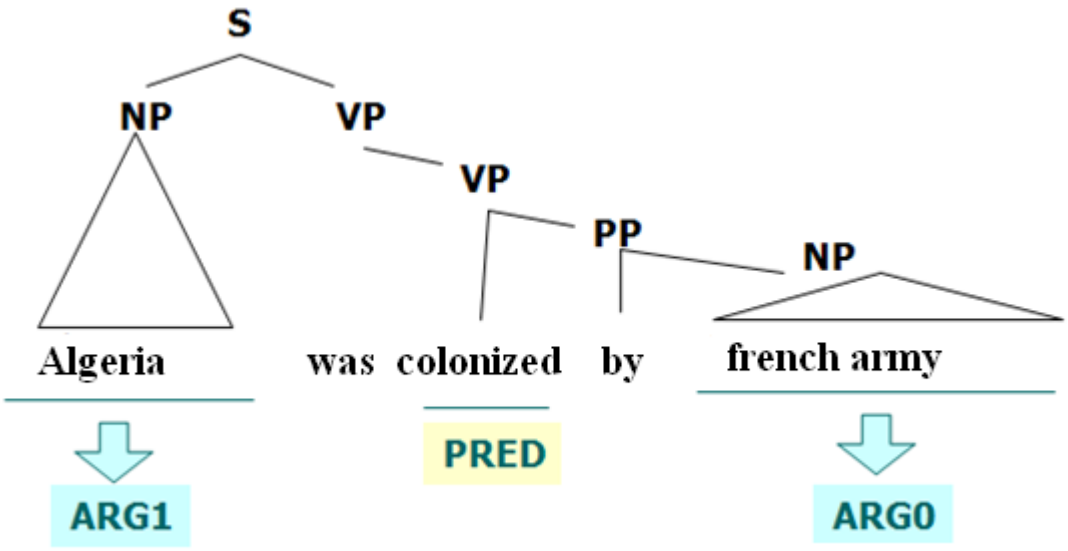

Figure 1.6 : Exemple de structure prédicat argument

5.4.2) Les différents niveaux d'inférence textuelle

Dans cette section nous allons présenter les différents niveaux d'inférences (Lexical, lexico syntaxique, sémantique (logique) et connaissance du monde) utilisées pour la détection de l'inférence textuelle.

5.4.2.1) L'inférence au niveau lexical

A ce niveau, l'inférence entre deux segments de textes est accepté s'il existe des mots semblables entre T et H, où les mots contenus dans la phrase H peuvent être inférés de T après des transformations lexicales (vanderwede et al., 2005). Les trois techniques d'inférence sont ci-dessous :

A) Les dérivations morphologiques

Ce mécanisme d'inférence considère que deux des termes sont équivalents si l'un peut être obtenu de l'autre après une dérivation morphologique. Il existe trois type de dérivations morphologiques :

- **La normalisation**

Exemple :
T : « l'**acquisition** d'un AIRBUS A380 par le roi FAHD ».
H : « le roi FAHD **a acquis** un AIRBUS A380 ».

La transformation <***d'acquisition***> en <***a acquis*** **>** a permis de faire la déduction de l'inférence entre les deux textes.

- **La dérivation nominale**

Exemple

T : Le GIA a donne de la **terreur** au peuple algérien.

H : Le GIA est un groupe **terroriste.**

La transformation de **terreur** en **terroriste** a permis de faire la déduction de l'inférence entre les deux textes.

- **Les relations entre noms et verbes**

Exemple

T : Mark gagne à tous les coups.

H : Mark est un gagnant.

La transformation de **Mark est un gagnant** en **Mark gagne** a permis de faire la déduction de l'inférence entre les deux textes.

B) Les relations ontologiques

Une ontologie est un ensemble structuré de concepts permettant de donner un sens aux informations. Elle est aussi un modèle de données qui représente un ensemble de concepts dans un domaine et les rapports entre ces concepts (Bourigault, 2004). Elle est employée pour raisonner au sujet des objets dans ce domaine.
Les concepts sont organisés dans un graphe dont les relations peuvent être : des relations sémantiques et des relations de subsomption.
L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné.
Ce mécanisme d'inférence se réfère à la relation **ontologique** qui existe entre deux termes. Ces différentes relations sont citées ci dessous.

- **La synonymie**

Représente un ensemble de mots interchangeables dans un contexte donné. Elle est souvent utilisée pour reconnaître l'inférence.

Exemple

T : « Jane a **abattue** Mark ».

H : « Jane a **tué** Mark ».

Autre exemple comme (''commencer''/''démarrer''), (''enlever ''/'' retirer'').

- **La généralisation (hypernymie)**

La relation d'Hypernymie est le terme générique utilisé pour désigner une classe englobant des instances de classes plus spécifiques. Y est un hypernyme de X si X est un type de Y.

Exemple

T : « On a coupé le **sapin** ».

H : « On a coupé **l'arbre** ».

La relation entre l'arbre et le sapin (l'arbre est une généralisation sapin) a permis l'inférence entre les deux textes.

- **L'hyponymie**

La relation Hyponymie est le terme spécifique utilisé pour désigner un membre d'une classe (relation inverse de Hypernymie). X est un hyponyme de Y si X est un type de Y.

Exemple

T : John a pris un **moyen de transport pour terrestre** pour faire le trajet Toulouse paris.

H : John a fait Toulouse Paris en **TGV**.

La relation entre **moyen de transport pour terrestre** et **TGV** qui a permis l'inférence entre les deux textes.

- **La relation de Méronymie**

X est un méronyme de Y si X est une partie de Y.

Exemple :

{Avion} a comme méronyme {{porte}, {moteur}} ;

C) La connaissance du monde dans l'analyse lexique

Ce mécanisme d'inférence se réfère à la connaissance du monde pour détecter l'inférence au niveau lexical (Len Schubert, 2002).

Exemple :

''Taliban ➔ organisation ''et ''yahoo → moteur de recherche ''

5.4.2.2) L'inférence au niveau lexico syntaxique

Au niveau lexico syntaxique l'hypothèse est représentée par des relations de dépendances syntaxiques.
La relation d'inférence entre T et H est définit comme un recouvrement des relations de H par les relations de T, ou le recouvrement est obtenu après une séquence de transformation appliquée à la relation de T. Les différents s types de transformations sont spécifies par :

A) Les transformations syntaxiques

Dans ce mécanisme d'inférence, la transformation se fait entre les structures syntaxiques qui ont les mêmes éléments lexicaux et préservent le sens de la relation entre elles (Vanderwende et al..,2005).
Ce genre de mécanisme inclut la transformation passive active et l'apposition[4].
Exemple :
« Mon chat, ce gentil petit siamois, est assis sur cette table ». « Il peut devenir : Mon chat est assis sur cette table, ce gentil petit siamois ! ».

B) L'inférence basée sur les paraphrases

Dans ce mécanisme d'inférence, la transformation modifie la structure syntaxique du segment du texte et quelques éléments lexicaux, mais elle garde la relation d'inférence entre le segment de texte original et celui qui est transformé.
Ce type de relation entre les deux segments est appelé dans la littérature « Paraphrase ». Des méthodes pour effectuer la transformation sont proposées dans (Lin et Pantel, 2001).
Exemple :
T : « Ce médicament est commercialisé au Canada seulement ».
H **:** « La commercialisation de ce médicament s'est effectuée au Canada seulement ».

C) La coréférence

La relation de coréférence met en relation un pronom et un antécédent éloigné l'un de l'autre dans la phrase. Par exemple :
« **L'Italie et l'Allemagne** ont tous deux joué deux matchs, **ils** n'ont perdu aucun match encore ».
Infère à
« Ni l'Italie ni l'Allemagne n'a encore perdu un match », cela inclut la transformation de coréférence **« ils ➔ l'Italie et l'Allemagne ».**

[4] L'apposition est une construction grammaticale dans laquelle deux éléments, normalement substantif expressions, sont placés à côté de l'autre, avec un élément servant à définir ou modifier les autres.. Lorsque ce dispositif est utilisé, les deux éléments sont censés être à l'apposition. Par exemple, dans l'expression "mon ami Alice" le nom "Alice" est à l'apposition de "mon ami".

5.4.2.3) L'inférence sémantique (logique)

A ce niveau, l'inférence entre deux segments de textes est acceptée si le sens des deux phrases se concorde. En d'autre termes, l'inférence textuelle est considérée comme un problème d'implication logique entre les sens des deux phrases (Tatu et al., 2006).
Pour cela, la structure prédicat argument est souvent utilisée, c'est-à-dire que, les segments de textes T et H sont transformés en prédicat et à travers des déductions logiques comme par exemple l'utilisation de la (preuve par réfutation[5]) on arrive à déduire l'inférence.
Un exemple des systèmes utilisant cette méthode d'inférence est décrit dans la section (**5.5.4.2**).

5.4.3) Les ressources utilisées

Dans les différents techniques d'inférence textuelle plusieurs ressources sont utilisées (WordNet, framnet, Cyc…). L'ensemble constitue un « écosystème » complet couvrant des aspects lexicaux, syntaxiques et sémantiques. Combinées, ces ressources fournissent un point de départ intéressant pour des développements sémantiques en TAL ou dans le cadre du Web sémantique, telle que la recherche d'information, l'inférence pour la compréhension automatique de textes, la désambiguïsation lexicale, la résolution d'anaphore et aussi l'inférence textuelle. Dans ce qui suit, nous allons définir les différentes ressources existantes et utilisées pour détecter l'inférence textuelle.

5.4.3.1) Le WordNet

WordNet (Miller, 1995) est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui est fondé sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991. Son but est de répertorier de classifier et de mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Le système se présente sous la forme d'une base de données électronique (Chaumartin, 2007).

Le synset (ensemble de synonymes) est la composante atomique sur laquelle repose WordNet. Un synset correspond à un groupe de mots, dénotant un sens ou un usage particulier. Un synset est défini par les relations qu'il entretient avec les sens voisins. Les noms et verbes sont organisés en hiérarchies. Des relations d'hyperonymie et d'hyponymie relient les « ancêtres » des noms et des verbes avec leurs «spécialisations». Au niveau racine, ces hiérarchies sont organisées en types de base.

À l'instar d'un dictionnaire traditionnel, WordNet offre ainsi, pour chaque mot, une liste de synsets correspondant à toutes ses acceptions répertoriées. Mais les synsets ont également d'autres usages : ils peuvent représenter des concepts plus abstraits, de plus haut niveau que les mots et leurs sens, qu'on peut organiser sous forme d'ontologie. Nous pouvons ainsi interroger le système quant aux hyperonymes d'un mot particulier. À partir par exemple du

[5] La réfutation est un procédé logique consistant à prouver la fausseté ou l'insuffisance d'une proposition ou d'un argument.

sens le plus commun du nom "car" (correspondant au synset "1. car, auto..."), la relation d'hyperonymie définit un arbre de concepts de plus en plus généraux:

```
1. car, auto, automobile, machine, motorcar
    => motor vehicle, automotive vehicle
        => vehicle
            => conveyance, transport
                => instrumentality, instrumentation
                    => artifact, artefact
                        => object, physical object
                            => entity, something
```

Dans cet exemple, il est clair que le dernier concept, "entity, something", est le plus général, le plus abstrait (il pourrait ainsi être le super-concept d'une multitude de concepts plus spécialisés).

Nous pouvons également interroger le système quant à la relation inverse de l'hypernymie, l'hyponymie. WordNet offre en fait une multitude d'autres ontologies, faisant usage de relations sémantiques plus spécialisées et restrictives. Nous pouvons ainsi interroger le système quant aux méronymes d'un mot ou d'un concept, les parties constitutives d'un objet ("HAS-PART"). Les méronymes associés au sens "car, auto..." du mot "car" sont :

```
1. car, auto, automobile, machine, motorcar
     HAS PART: accelerator, accelerator pedal, gas pedal, gas,
               throttle, gun
     HAS PART: air bag
     HAS PART: auto accessory
     HAS PART: automobile engine
     HAS PART: automobile horn, car horn, motor horn, horn
     (...)
```

5.4.3.2) Le FrameNet

FrameNet (Baker, Fillmore et Lowe, 1998), projet mené à Berkeley à l'initiative de Charles Fillmore, est fondé sur la sémantique des cadres (frame semantics). FrameNet a pour objectif de documenter la combinatoire syntaxique et sémantique pour chacun des sens d'une entrée lexicale à travers une annotation manuelle d'exemples choisis dans des corpus sur des critères de représentativité lexicographique. Les annotations sont ensuite synthétisées dans des tables, qui résument pour chaque mot les cadres avec leurs arguments syntaxiques.

5.4.3.3) Le Cyc

Cyc est un projet d'Intelligence Artificielle lancé en 1984 par Doug Lenat. Cyc vise à regrouper une ontologie et une base de données complètes sur le sens commun, pour permettre à des applications d'intélligence artificielle. D'effectuer des raisonnements similaires à ceux des humains. Des fragments de connaissances typiques sont par exemple : « les chats ont quatre pattes » ; « *Paris est la capitale de la France* ». Elles contiennent des

termes (PARIS, FRANCE, CHAT?) et des assertions (« *Paris est la capitale de la France* ») qui relient ces termes entre eux. Grâce au moteur d'inférence fourni avec la base Cyc, il est possible d'obtenir une réponse à une question comme « Quelle est la capitale de la France ? » La base Cyc contient des millions d'assertions (faits et règles) rentrées à la main.

5.5) L'analyse des systèmes participant au RTE 2

Nous avons marqués pour chaque groupe de recherche participant au RTE2 les types d'inférences utilisés. Les résultats sont affiches dans le tableau 1.6.

Type d'analyse / Groupes de recherches	lexicale	syntaxique	lexico-sémantique	Logique	numérique	Temporelle
UNED	+	+	+		+	
UMESS	+					
MITRE	+			+		
IRST	+	+				
GOGEX	+		+	+		
LCC'S	+		+			
C&C	+	+				

Tableau 1.1 Représentation des différents types d'inférences entrepris par les groupes de recherches

5.5.4) Quelques exemples d'inférence utilisés par des groupes de recherches

Dans le RTE 2 nous avons remarqué que tous les groupes de recherches n'ont pas utilisé d'inférence temporelle dans leurs systèmes et à l'heure actuelle, les résultats du RTE 3 ne sont pas encore publiés officiellement mais d'après notre lecture des différentes publications des groupes de recherches participant au RTE3, il y a deux groupes qui ont fait allusion à l'inférence temporelle. Pour cela, nous avons choisi de décrire leurs systèmes.

5.5.4.1) La reconnaissance de l'inférence textuelle basée sur l'analyse de dépendance et WordNet (Université nationale de l'éducation a distance de Madrid)

Le système présenté montre comment des informations sémantiques peuvent être extraites du texte en utilisant les structurations syntaxiques données par l'analyse de dépendance, et des ressources lexico- sémantiques comme Word Net peuvent développer le RTE.

Les techniques utilisées par ce système sont les suivantes :

- l'analyse dépendance du texte et de l'hypothèse.
- l'inférence lexicale entre les nœuds des arbres en utilisant Word Net.
- la concordance entre les arbres de dépendance basée sur la notion de l'inclusion.

A) L'architecture du système

L'architecture du système est montrée dans la figure suivante (**Figure 1.7**) :

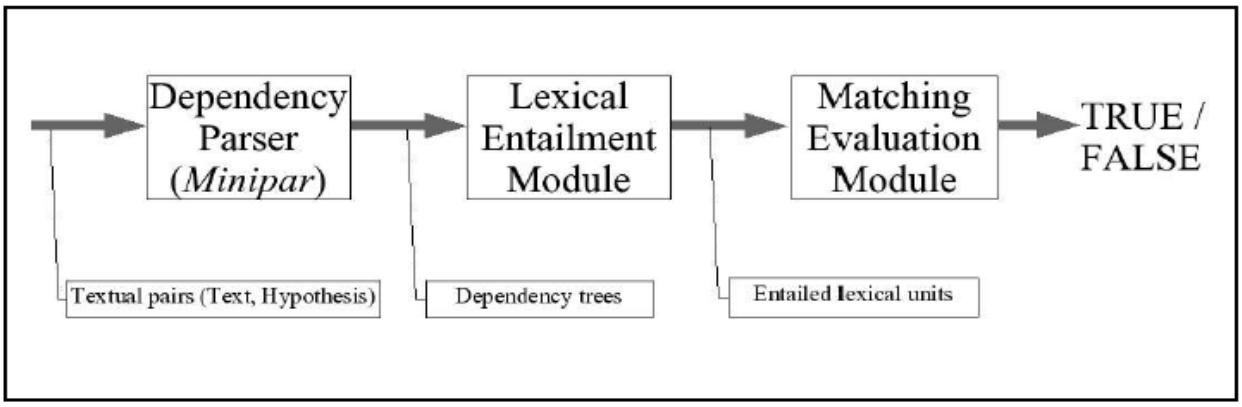

Figure 1.7 : L'architecture du système

Cette architecture est composée de Trois modules :

- **L'analyse de dépendance :** Elle consiste à normaliser les informations du dataset, de générer les dépendances existantes entre les mots et de donner à la sortie un arbre de dépendance constitué de nœuds qui représentent les mots de la phrase et d'arcs qui représentent les dépendances entre les nœuds. Ce travail est réalisé par un logiciel nommé « Lin's Minipar ».

- **L'analyse lexicale :** prend les informations données par l'analyse de dépendance et retourne les mots de l'hypothèse H qui sont infères du texte T. Ce module utilise WordNet pour détecter les relations de (synonymie, hyponymie, meronymie) entre les unîtes lexicales.

- **Les relations entre les arbres de dépendance** : le but est de déduire si l'arbre de l'hypothèse est recouvert par l'arbre de dépendance du texte, Pour cela, la règle établie est qu'un arc est dit recouvert s'il est dans le même emplacement que dans l'arbre représentant le texte et il y a une inférence entre ces nœuds et celle du texte. La figure ci-dessous (figure 1.8) reprend ce genre de recouvrement.

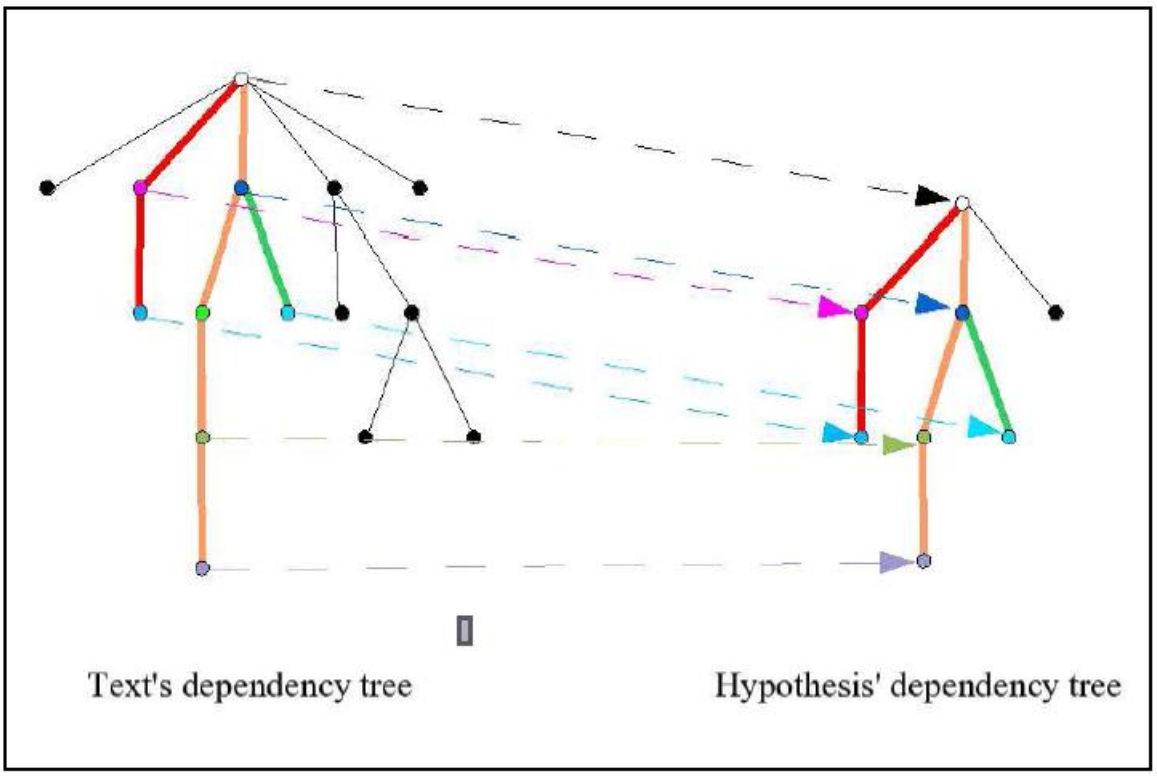

Figure 1.8: Exemple de recouvrement entre arbre de dépendance

B) L'expérimentation du système

Le groupe a soumi deux systèmes au challenge.

- **Système 1**

Le systeme1 n'utilise que les deux premiers modules, et la décision de l'existence d'inférence est prise par rapport au nombre de nœuds de l'hypothèse infère de l'arbre de dépendance du texte.

- **Système 2**

Le système 2 utilise les 3 modules et la décision est prise par rapport au nombre d'arc recouverts.

Les résultats sont affiches dans le **tableau 1.2**. L'utilisation de WordNet seule a donné de bons résultats, mais en ajoutant le module de recouvrement il décroît les performances du système.

Les systèmes	Précision
Système 1 :	56,37 %
Système 2 :	54,75 %

Tableau 1.2: Les valeurs de précision des systèmes

La notion de recouvrement n'est pas appropriée pour le RTE, car un large recouvrement n'implique pas une inférence sémantique, et un faible recouvrement n'implique pas une différence sémantique. L'utilisation de Word Net a contribué à l'inférence au niveau lexical et a augmenté les performances du système. Dans cette direction, les prochaines étapes seront de reconnaître et d'évaluer les inférences entre les expressions numériques, **les entités nommées** [6]et les expressions temporelles.

C) L'évolution du système

Ce qui a été développé pour le RTE2 est un module pour la détection des expressions numériques, ce qui a permis d'augmenter fortement **la précision** (harrera et al.,2006). La figure suivante montre comment le module est introduit dans leur système.

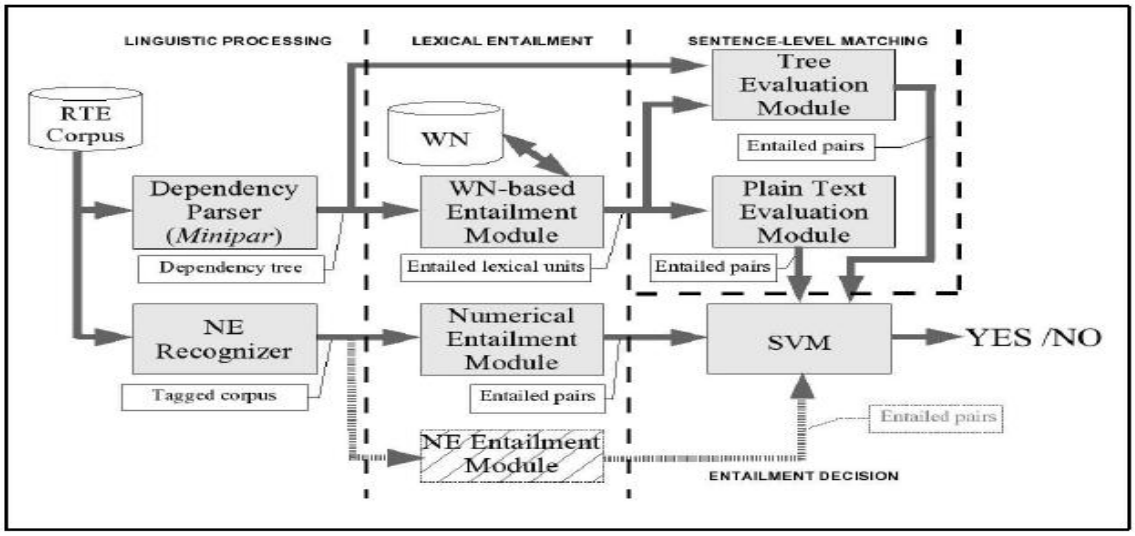

Figure 1.9: Architecture du système UNED

Dans le RTE 3, le groupe s'est focalisé sur l'inférence entre les entités nommées. Il a défini les relations d'inférences entre les entités nommées (Rodrigo et al., 2007). Exemple :

- Nom propre E1 infère nom propre E2 si une chaîne E1 contient la chaîne E2.
- une expression du temps t1 infère une expression du temps T2 si l'intervalle de temps exprimée dans t1 est inclus dans l'intervalle T2.

Ce module de d'inférence a lui aussi contribué à augmenter la précision (Rodrigo et al, 2007).

5.5.4.2) COGEX (université du Texas, USA)

Le système utilise une approche logique pour résoudre l'inférence textuelle. En d'autres termes, l'inférence textuelle est considérée comme un problème d'implication logique entre les sens des deux phrases (Tatu et al., 2006).
La description du système et l'évolution qui s'est produite dans chaque challenge est décrite dans ce qui suit.

[6] Les entités nommées désignent l'ensemble des noms de personnes, de lieux, d'entreprise contenues dans un texte.

A) La description du système

La première étape consiste à transformer le texte et l'hypothèse en forme logique (Moldovan and Rus, 2001).

Pour cela il faut d'abord transformer du langage nature a un format prédicat argument, pour cella le groupe utilise WordNet pour lier le prédicat avec ses argument. Concrètement WordNet produit des relations entre les synsets, et chaque synset lui correspond un prédicat.

Le prédicat peut avoir un ou plusieurs arguments et le prédicat qui correspond au nom a un seul argument en général, et le prédicat qui correspond à un verbe a trois arguments : l'événement, le sujet et le complément d'objet.
Pour chaque relation dans la chaîne lexicale[7], le système génère un axiome utilisant les prédicats qui correspondent au synset de la relation.

Par exemple : il y a une relation d'inférence entre le verbe **vendre** et le verbe **payer.**
Le système génère l'axiome suivant pour cette relation :
Vendre_VB_1(e1,x1,x2) ➔ payer_VB_1(e1,x1,x3)
Ce type d'axiome contribue à l'inférence quand une chaîne lexicale est trouvée.

Apres la transformation des deux paires de texte en format logique le groupe utilise la preuve par « l'absurde » ou ''preuve par contradiction'' (Wos, 1998). La négation de l'hypothèse H est réalisée s'il y a une contradiction ou une déduction de contradiction par rapport au texte T, nous concluons que l'hyponyme est dérivable du texte.

B) L'évolution du système

Il a été développé pour le RTE 2 un module qui traite la négation dans la transformation du texte en prédicat et un autre module qui fait une analyse sémantique en tant que pré traitement pour donner les relations existantes entre le verbe et ses arguments et aussi entre les arguments eux- mêmes (Tatu et al.,2006).

Pour le RTE3 le groupe a développé et intégrer a leur système plusieurs outils.
Dans ce qui suit nous allons présenter l'architecture du système et les nouveaux outils conçus et utilises pour améliorer l'inférence.
Le schéma du dernier système conçu pour le RTE 3 par le groupe est donné par la figure ci-dessous.

[7] Une chaîne lexicale est une chaîne où il y a une relation entre deux synsets.

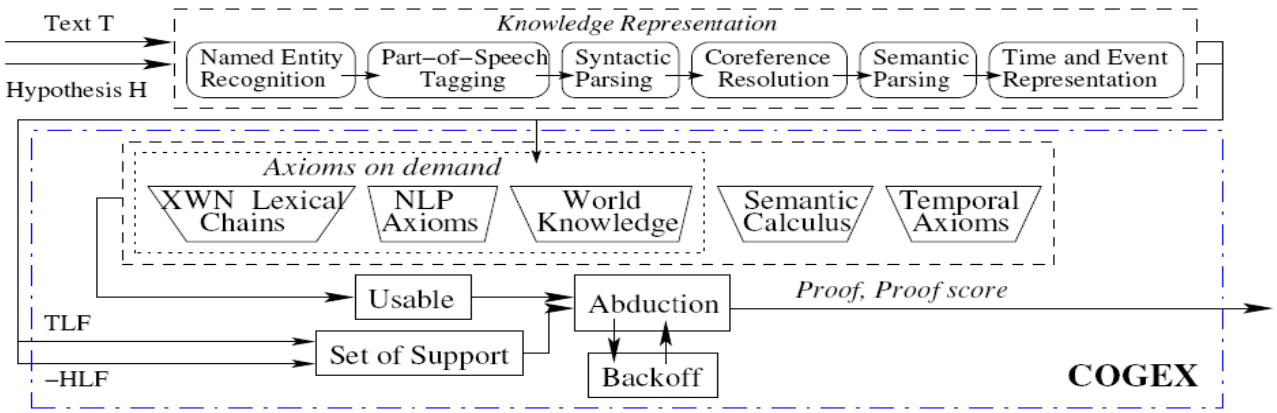

Figure 1.10 : Architecture du système

- **EXtended WordNet**

XWN (eXtended WordNet) est un projet qui a pour but d'enrichir les relations du dictionnaire WordNet avec des relations sémantique entre les synsets et les transforment en format logique (Tatu et Moldovan, 2007).

- **TARSQI**

C'est un système modulaire pour l'annotation automatique temporelle qui ajoute les expressions du temps, des événements et des relations temporelles de l'actualité des textes (Venhaguane et al. ,2005).

- **Outil pour la gestion des coréférences**

Pour relier les phrases dans les textes longs et, résoudre le problème qui est apporté par les coréférences dans l'inférence textuelle, l'outil développé combine l'algorithme Hobbs (Hobbs, 1978) et l'algorithme de résolution d'anaphore (Lappin and Leass, 1994).
Pour le RTE, il est important d'avoir les relations entre les prédicats d'un long texte.

Exemple 1 : *George Bush grandit à Greenwich au Connecticut, Il est à l'époque membre d'une confrérie étudiante secrète devenue célèbre*.
Lier George Bush et *il,* est une des taches que l'outil doit résoudre.

Le développement du XWN-KB a eu un impact considérable sur le RTE, mais l'utilisation du TARSQI n'a donné aucun impact sur le résultat car l'utilisation des expressions temporelles dans ce corpus est inexistante.

5.5.5) Conclusion

Dans les travaux entamés par UNED sur les entités nommées, le groupe a établi plusieurs règles d'inférence entre les entités nommées, parmi lesquelles se trouve une règle d'inférence entre les expressions temporelles. Celle-ci peut être considérer comme une contribution implicite à l'inférence temporelle. Mais concrètement l'inférence temporelle est considérée comme une perspective pour leurs prochaines recherches.

5.6) Conclusion

Dans ce chapitre nous avons explorés l'apport du RTE dans les différentes applications du TALN (RI, QR, EI et RA) et nous avons exploré les différentes approches utilisées pour détecter l'inférence (lexical, lexico syntaxique, sémantique et logique). Aussi nous avons analysé les approches des différents groupes de recherches qui ont participe au challenge Pascal RTE. Cette étape nous a permis de découvrir les chemins qui n'ont pas encore été pris pour détecter l'inférence textuelle.

Enfin nous nous sommes focalisés à décrire les systèmes qui ont mentionné l'aspect temporel dans leurs recherches. Nous avons remarqué que dans les trois RTE qui se sont déroulés, l'inférence temporelle est une perspective qui n'est pas encore entamée. Nous allons justement décrire dans le prochain chapitre l'aspect temporel dans le RTE.