Sous-langage d'application et LTAG : le système EGAL

Patrice Lopez, Christine Fay-Varnier, Azim Roussanaly

▶ **To cite this version:**

Patrice Lopez, Christine Fay-Varnier, Azim Roussanaly. Sous-langage d'application et LTAG : le système EGAL. 6ème Conférence annuelle sur le Traitement Automatique du Langage Naturel - TALN'99, Jul 1999, Cargèse, Corse, France, 10 p. inria-00098780

HAL Id: inria-00098780

https://inria.hal.science/inria-00098780

Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sous-langages d'application et LTAG : le système EGAL

Patrice Lopez, Christine Fay-Varnier et Azim Roussanaly

LORIA
BP 239 - 54506 Vandœuvre-lès-Nancy, France
{lopez,fay,azim}@loria.fr

Résumé

Nous présentons un système dédié à la conception et au test d'un sous-language d'application pour un système de Dialogue Homme-Machine. EGAL se base sur une grammaire LTAG générale de la langue qui est spécialisée à une application donnée à l'aide d'un corpus d'entraînement. Un double effort a porté premièrement sur la définition d'une méthodologie précise passant par une expérimentation de type Magicien d'Oz pour le recueil des corpus et des estimations de la représentativité du corpus de conception, et, deuxièmement, sur la spécification des composants du système en vue de mettre en œuvre des outils convivaux, génériques et ouverts.

1. Introduction

1.1. Motivations

Suite au développement ces dernières années de grammaires lexicalisées à large couverture pour l'écrit, et dans la mesure où on s'intéresse au Dialogue Homme Machine finalisé, il est intéressant d'étudier comment interpréter la langue orale spontanée à partir de ce style de grammaires et de techniques additionnelles dépendant notamment du domaine d'application. Aux niveaux lexical et syntaxique, cette adaptation passe à notre sens par les points suivants :

- modéliser les phénomènes oraux jugés agrammaticaux ou peu fréquents pour l'écrit mais qui présentent une régularité importante pour la langue spontanée, en particulier les incises et les ellipses (Price *et al.*, 1989) ;
- mettre en œuvre des techniques robustes d'analyse afin d'être tolérant au caractère peu normalisé de la syntaxe de l'oral (van Noord *et al.*, 1998) ;
- spécialiser un lexique et une grammaire motivés linguistiquement et destinés à du texte tout venant pour un type de dialogue et un domaine particulier.

Nous nous intéressons ici au dernier point dont la difficulté justifie parfois l'abandon de modèles linguistiques conçus manuellement pour des analyses plus superficielles et des modèles construits selon des techniques purement probabilistes (Bod, 1995). Il faut cependant bien noter qu'une grammaire écrite manuellement suivant des principes linguistiques permet une compréhension plus précise des phénomènes mis en jeu. En particulier, une telle grammaire permet de

prendre en compte l'importante ambiguïté du niveau syntaxique qui constitue l'une des différences essentielles entre le langage naturel que nous souhaitons traiter et les langages pseudo-naturels qui sont bien souvent ceux pris en compte. D'autre part, les systèmes probabilistes nécessitent des corpus d'entraînement très volumineux dont la conception peut relever d'un effort tout aussi important que l'écriture manuelle d'une grammaire.

Nous présentons une méthodologie et un système nommé EGAL (Extraction de Grammaire d'Arbres Lexicalisés), capable d'extraire de façon semi-automatique un sous-langage applicatif à partir d'une grammaire générale et de corpus de type Magicien d'Oz. Une fois la sous-grammaire d'application obtenue, un module d'analyse permet de la mettre à l'épreuve sur un corpus de test suivant plusieurs algorithmes et stratégies. Les dérivations partielles et complètes peuvent alors être visualisées et comparées en considérant différents critères. Ces tests permettent en outre d'obtenir des informations quant à la représentativité du corpus initial utilisé pour décrire le sous-langage. Enfin, grammaire et analyseur sont destinés à être intégrés dans des systèmes de dialogue opérationnels.

Notre objectif est de concevoir des interfaces vocales Homme-Machine génériques et portables permettant la communication en langue spontanée. Pour illustrer la méthodologie et le système proposés, nous avons choisi une application cible dont nous avons collecté des corpus expérimentaux, extrait un lexique et une grammaire d'application et enfin testé différentes stratégies d'analyse.

1.2. Grammaires d'Arbres Adjoints Lexicalisées

La lexicalisation d'un formalisme syntaxique consiste à associer à chaque entrée lexicale une modélisation des contextes syntaxiques dans lesquels elle peut être utilisée. Lexique et grammaire se confondent alors en un lexique syntaxique. L'intérêt de la lexicalisation est double : tout d'abord, la possibilité d'associer à chaque entrée lexicale un degré de finesse des descriptions particulièrement souple, évitant les effets de bords de grammaire fondée sur une régularité de la langue (bien souvent illusoire dès que l'on traite de la langue naturelle même finalisée). Ensuite, cette propriété permet d'appliquer des heuristiques d'analyse, renvoyant de nombreux problèmes d'ambiguïtés à des ambiguïtés lexicales plus simples à traiter.

Le choix du formalisme est essentiel pour la représentation et la compréhension des phénomènes linguistiques. Il l'est aussi pour son application effective au traitement automatique de la langue. Nous avons choisi les Grammaires d'Arbres Adjoints Lexicalisées (LTAG), formalisme syntaxique intéressant pour l'analyse comme la génération de langage naturel, en particulier grâce à sa propriété de lexicalisation et à un domaine de localité étendu. Des études linguistiques et des développements de grammaires à large couverture notamment pour l'anglais et le français ont permis d'exploiter ces propriétés. De plus des modèles probabilistes fondés sur les LTAG comme les TAG stochastiques (Schabes, 1992) ou le super-tagging (Srinivas, 1997), permettent d'optimiser le traitement des ambiguïtés lexicales et syntaxiques par des choix préférenciels.

La lexicalisation présente cependant certaines contreparties, en particulier la tâche de conception d'une grammaire. Encore en cours d'amélioration, la grammaire anglaise du système XTAG (Doran *et al.*, 1994) a d'ors et déjà demandé plus de sept ans de développement, celle du français (Abeillé *et al.*, 1994) plus de cinq ans. Une grammaire à large couverture peut contenir plus d'un millier de motifs d'arbres élémentaires non instanciés, appelés ici *schèmes* (Candito, 1999), et nécessite la conception d'une base de données syntaxiques décrivant pour chaque lemme les

arbres ou familles d'arbres correspondants. En considérant une application donnée, utiliser une grammaire complète de la langue aboutirait à un nombre prohibitif d'hypothèses. D'autre part notre but est d'éviter d'avoir à concevoir entièrement une nouvelle grammaire spécifique pour chaque application.

Des travaux sur l'utilisation de LTAG dans des systèmes de dialogue ont été récemment menés (Roussel & Halber, 1997)(Lopez, 1998a) mais l'adaptation d'une grammaire générique à une application spécifique reste un problème essentiel pour une utilisation pratique de ce formalisme dans des interfaces vocales. De plus expérimenter des algorithmes d'analyse et des heuristiques particulières nécessite des outils génériques et ouverts.

2. Méthodologie de recueil

2.1. Sous-langage d'un système de dialogue

Un sous-langage se définit comme un ensemble d'énoncés liés par un sujet limité, utilisés pour une fonction particulière et engendrés par une grammaire et un vocabulaire spécifiques (Deville, 1989). Deux facteurs viennent restreindre le langage général dans un système de Dialogue Homme-Machine : d'une part le type de dialogue finalisé qui est mis en œuvre (dialogue de commande, d'assistance, ...) et d'autre part le domaine d'application du système. Un sous-langage n'est pas un simple sous-ensemble du langage complet : une application peut nécessiter l'usage des termes techniques ne relevant que du domaine. Il est donc important qu'un système propose des méthodes pour décrire de nouveaux mots et de nouveaux contextes syntaxiques afin de prendre en compte les structures échappant à une grammaire générale de l'écrit.

Du point de vue opératoire, outre la diminution de la combinatoire, une grammaire restreinte à un sous-langage rend également réaliste la possibilité de mettre en œuvre une grammaire écrite manuellement, chose nettement plus difficile par exemple pour des systèmes de dictée.

En principe le système n'a pas à comprendre des mots hors du sous-langage. Ceci ne signifie pas un comportement déterministe impliquant l'arrêt d'une analyse en cas d'incomplétude de la grammaire ou d'agrammaticalité, ou encore un glissement d'un énoncé hors de la grammaire vers un énoncé grammatical par effet de bord par exemple d'une analyse probabiliste. L'usage d'un mot ou d'une structure hors de la grammaire doit être détecté en tant que tel et entraîner éventuellement des intéractions verbales supplémentaires avec l'utilisateur.

2.2. Expérimentations de type Magicien d'Oz

Une expérience de Magicien d'Oz consiste à simuler le fonctionnement d'un système de dialogue à l'insu de l'utilisateur, en vue d'établir un recueil des intéractions possibles pour une application donnée. Le corpus obtenu, auquel on peut attribuer une représentativité subjective, devient alors une source de travail notamment pour la modélisation linguistique des énoncés en vue de la compréhension et de l'analyse du dialogue.

Un problème dans la conception de ce type de corpus réside dans la représentativité du recueil par rapport au sous-langage que l'on souhaite décrire. Si le principe de sous-langage est fondé, on peut considérer qu'à partir d'une certaine taille du corpus l'augmentation de la taille du vocabulaire et de la grammaire n'évoluera plus significativement. Une méthodologie d'estimation de la taille du corpus à considérer pour atteindre une certaine représentativité est une chose *a priori* importante sinon nécessaire pour la mise en œuvre pratique d'expérimentations de Magicien d'Oz.

Notre démarche consiste à recueillir un corpus qui sera par la suite classiquement divisé en deux parties, la première permettant de concevoir une grammaire du sous-langage (*corpus de conception ou d'entraînement*) et la seconde dédiée aux tests (*corpus de test*).

Nous avons évoqué différents aspects essentiels au type de système que nous mettons en œuvre : démarche expérimentale de type Magicien d'Oz afin de recueillir des corpus, spécialisation/conception d'une grammaire lexicalisée à partir de ces derniers dédiée à l'analyse de l'oral, test de la grammaire et évaluation de la représentativité du corpus de conception. Nous examinons ici quatre ateliers de modèlisation linguistique relativement à ces aspects.

3. Positionnement par rapport aux systèmes existants

Le système XTAG (Doran *et al.*, 1994) propose une grammaire LTAG de l'anglais à large couverture, des modules de visualisation de la grammaire et de résultats d'analyse et un analyseur de type Earley (Schabes, 1994). Cependant trois principaux aspects nous semblent limitatifs :

- le système de conception de la grammaire est dédié à une grammaire générale de l'écrit et rien ne permet la restriction de cette grammaire à une sous-grammaire basée sur un corpus, ni une adaptation à l'analyse d'énoncés oraux ;
- l'analyseur fournit une réponse binaire (phrase acceptée ou refusée) difficilement compatible avec le test d'une grammaire de taille importante où on souhaiterait manipuler des résultats partiels et obtenir des diagnostics d'erreurs ;
- la conception d'un nouvel analyseur se justifie donc mais l'intégration au système XTAG de nouvelles composantes est hors de la compétence d'une personne n'ayant pas participé au développement du système. Plus généralement ce système n'a pas été conçu dans un souci de diffusion car il manipule des formats propres non spécifiés, nécessitant une expertise pointue pour son installation, etc...

L'atelier de "Génie linguistique" pour LFG du LIMSI (Briffault *et al.*, 1997) couvre les mêmes possibilités que le système précédent avec cependant une grammaire plus modeste. Les points forts essentiels de ce système sont l'intégration d'un niveau sémantique basé sur les graphes conceptuels et des choix techniques habiles permettant son évolution comme son intégration dans d'autres systèmes.

Le système GEPETTO (Ciravegna *et al.*, 1997) offre également un atelier de test incluant plusieurs algorithmes d'analyse syntaxiques. Ce système s'inscrit dans une démarche très générale, indépendante du formalisme. Cette généralité peut apparaître comme un avantage, mais entraîne une perte de fonctionnalité par rapport à des outils adaptés aux concepts et aux données d'un formalisme particulier. Si la méthodologie associée prévoit une division des corpus d'entraînement et de tests, il n'est pas prévu l'adaptation de ces outils dans d'autres applications, ni la spécialisation d'une grammaire générale.

Cette dernière constatation s'applique également pour l'environnement Hdrug (van Noord & Bouma, 1997), un outil utilisé à la fois pour le développement d'applications et pour l'expérimentation de stratégies d'analyse en paticulier pour l'oral. La grammaire utilisée est une grammaire HPSG simplifiée.

4. Présentation du système

L'organisation générale d'une grammaire lexicalisée destinée à l'analyse syntaxique repose sur trois sources de données :

- une base morpho-syntaxique qui, à une forme fléchie, associe un lemme, une catégorie syntaxique et un ensemble de traits morphologiques ;
- une base syntaxique qui, à un lemme donné, associe un ensemble d'arbres élémentaires représentant les contextes syntaxiques dans lequel ce lemme peut être utilisé ;
- enfin un ensemble de schèmes (Candito, 1999).

La composante de conception grammaticale du système que nous présentons ici s'articule autour de ces trois types de données (voir figure 1).

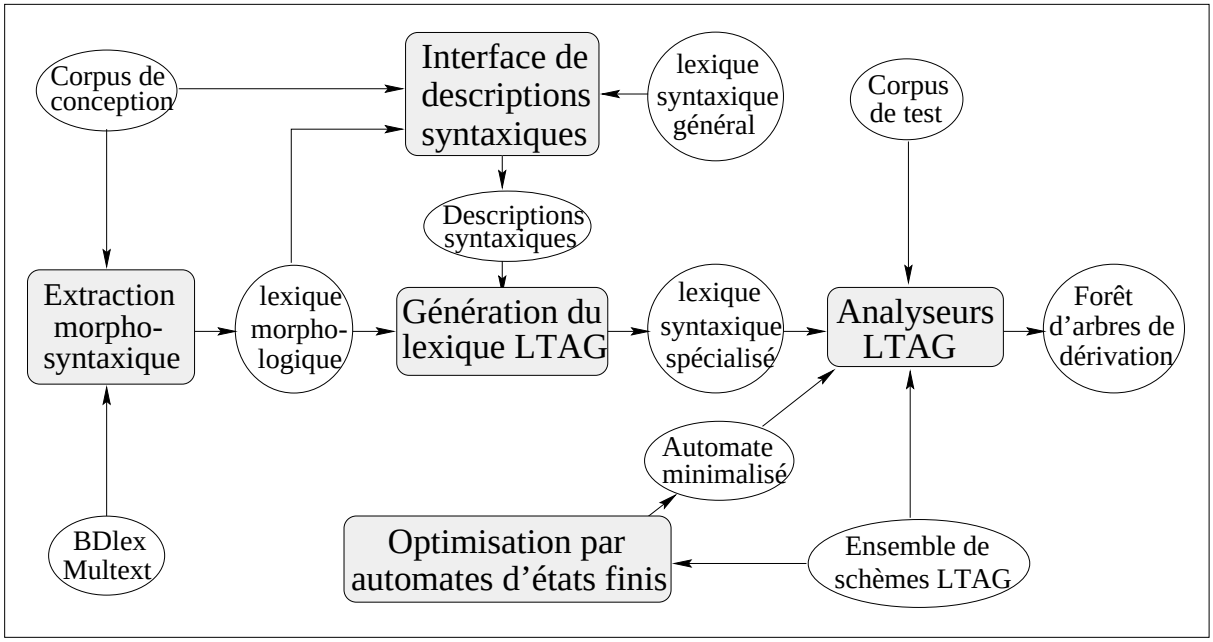

FIG. 1 – *Vue générale du système EGAL.*

4.1. Génération lexicale semi-automatique

Extraction morpho-syntaxique Étant donné un corpus d'entrainement, il s'agit simplement d'exploiter les bases morpho-syntaxiques existantes (Multext, BDlex) en extrayant automatiquement les informations nécessaires pour l'ensemble des mots employés.

Descriptions syntaxiques L'objectif de ce module est d'identifier les propriétés syntaxiques à associer à un lemme afin de sélectionner les structures syntaxiques dans lequel il peut s'employer. Cette identification est la seule nécessairement non-automatisable et se fait à l'aide d'une interface dédiée à des non-grammairiens.

L'idée est de proposer à un utilisateur une série de tests linguistiques et de questions illustrés par des exemples afin de caractériser un lemme (par exemple : le verbe peut-il s'appliquer sous une forme réflexive? ou avec quel auxiliaire s'emploie-t'il?). Les réponses peuvent se faire soit de manière complète en vue de caractériser un lemme selon tous ses emplois possibles, soit

en s'intégrant dans la méthodologie proposée, en vue de caractériser les seuls contextes apparaissant dans le corpus de conception (une extraction des énoncés portant une occurrence du lemme est proposée simultanément). Nous allons ici au delà de la conception classique d'un sous-langage en spécialisant les lemmes en terme de catégories selectionnées mais également en terme de contextes syntaxiques spécifiques. Nous ne nous appuyons donc pas sur le principe de famille d'arbres introduit par le système XTAG, essentiellement pour des raisons de performances : plutôt que de reléguer la sélection des bons arbres d'une famille à associer à une ancre donnée par unification au moment de l'instanciation, les bons arbres sont déjà déterminés à l'aide de ces descriptions et seront directement notés dans le lexique syntaxique.

Un outil complémentaire, à destination de linguistes cette fois, permet de concevoir les tests linguistiques. On peut noter que :

- ces descriptions sont indépendantes du formalisme lexicalisé qui est employé (LTAG ou HPSG par exemple) ;
- ce module permet d'intégrer de manière simple de nouveaux mots à un système, en caractérisant les formes fléchies non reconnues lors de la phase d'extraction.

Ensemble de schèmes Nous partons du principe que nous disposons d'un ensemble de schèmes d'arbres élémentaires, généré par exemple automatiquement à l'aide d'un système comme (Candito, 1996) duquel notre système se veut complémentaire. La série de tests du module précédent revient en particulier à déterminer la place d'un lemme dans la structuration proposée dans (Candito, 1999). Notons qu'un éditeur permet de concevoir des schémas d'arbre nouveaux et de les intégrer au sein d'une arborescence de familles d'arbres. D'autre part, un module d'optimisation par compaction d'automates d'états finis selon des techniques proches de (Evans & Weir, 1997) permet une factorisation des sous-structures communes présentées par l'ensemble des schèmes afin de rendre plus efficace l'analyse.

Génération automatique du lexique syntaxique spécialisé LTAG Cette phase consiste à produire le lexique syntaxique spécialisé en exploitant les informations des trois bases de données précédentes. Les liens vers les schèmes sont simplement notés par références externes.

4.2. Atelier de test d'analyse

Après avoir obtenu une grammaire pour un sous-langage d'application à partir d'un premier corpus, ce module a pour tâche de tester le résultat obtenu sur un second corpus. Ceci permet :

- de tester et comparer différentes heuristiques et stratégies d'analyse ;
- de visualiser[1] les résultats d'analyse complets et partiels afin d'améliorer la grammaire obtenue et d'étudier notamment les phénomènes agrammaticaux observés sur le corpus de test ;
- de donner des informations sur la représentativité du corpus initial qui a permis de concevoir la grammaire.

Cet atelier propose plusieurs algorithmes et heuristiques d'analyse :

- un algorithme ascendant de type CKY pour les LTAG avec le mécanisme de prédiction simple suggéré dans (Vijay-Shanker & Weir, 1993) ;

1. Nous utilisons dans EGAL une API d'édition d'arbres conçue par Rodrigo Reyes (Thomson LCR et TALaNa).

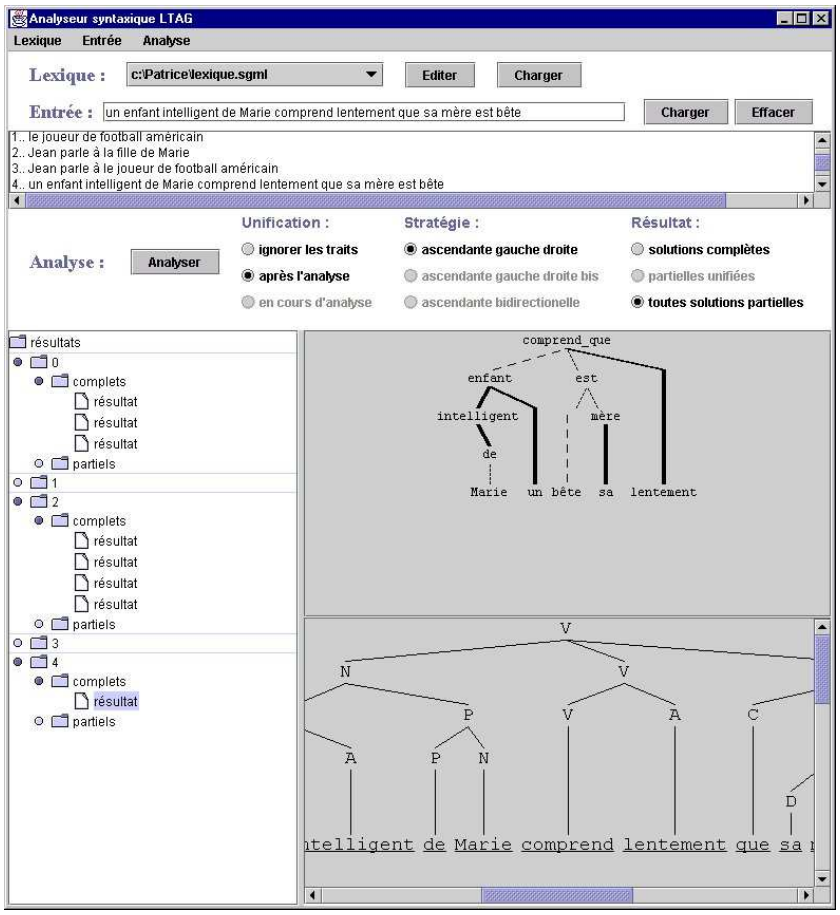

FIG. 2 – *Atelier de test d'analyse.*

– un algorithme ascendant par connexité (Lopez, 1998b) ;
– une implantation de l'algorithme de (Schabes, 1994) ascendant de type Earley.

Les analyseurs ascendants fournissent les analyses complètes et partielles avec ou sans unification des structures de traits utilisées en LTAG. Ces différents types de résultat ont pour but de permettre un véritable test de la grammaire utilisée en identifiant l'étape impliquée dans un échec de l'analyse.

4.3. Choix techniques

L'ensemble de l'implantation s'appuie sur Java essentiellement pour des raisons de portabilité qui faciliteront la diffusion de ces outils[2], et sur un codage systématique des données avec une application de XML nommé TagML (Tree adjoining grammar Markup Language). TagML permet de représenter, structurer et assurer la cohérence de l'ensemble des données et ressources nécessaires à l'exploitation d'une grammaire LTAG conformément à des DTD. TagML rend également possible le codage des redondances structurelles de la grammaire et celui des équations de traits partagées entre plusieurs schémas d'arbres élémentaires afin d'améliorer les étapes d'analyse et d'unification. Tout analyseur respectant cette norme en entrée et sortie pourra ainsi s'intégrer aisément au système.

2. Nous souhaitons rendre prochainement disponible l'ensemble des sources du système afi n de susciter dans la communauté d'éventuelles contributions.

5. Grammaire et analyse du corpus Gocad

5.1. Une application cible : Gocad

L'application Gocad a pour but la modélisation de surfaces géologiques. Le protocole et la manipulation de Magicien d'Oz employés sont présentés dans (Chapelier *et al.*, 1995). Ils nous ont permis de recueillir un corpus [3] d'énoncés présenté table 1.

nombre d'énoncés utilisateur	nombre de mots	nombre moyen de mots/énoncé.
862	5535	6,42

TAB. 1 – *Taille du corpus Gocad*

5.2. LTAG du sous-langage d'application

Le corpus obtenu a été divisé en corpus de conception (80% des énoncés) et de test (20% restant). La taille de la grammaire obtenue par le système EGAL est présentée table 2. Le nombre total de liens vers un schème donne une métrique de la taille totale du lexique syntaxique.

nombre de formee fléchies	nombre de schèmes	nombre de liens vers schème
526	71	1776

TAB. 2 – *Taille de la grammaire LTAG associé au corpus de conception Gocad*

5.3. Représentativité du corpus d'entraînement

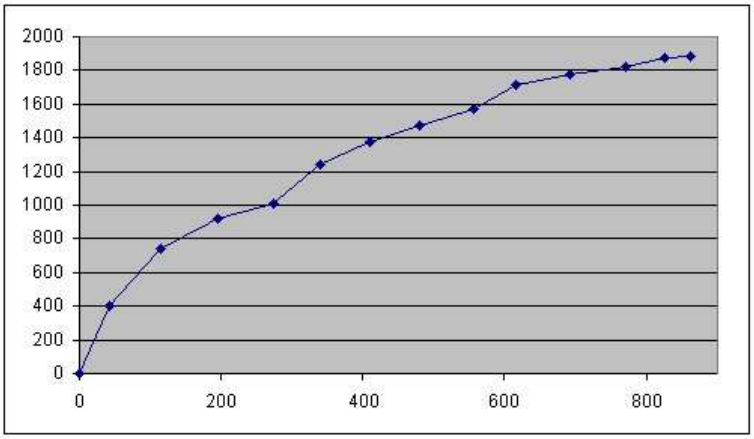

FIG. 3 – *Evolution de la taille de la grammaire LTAG générée en fonction de la taille du corpus.*

Les phases d'extraction et de génération du lexique syntaxique pour Gocad étant d'une durée faible (de l'ordre de 20 secondes pour chacune d'elle), il est possible de réaliser des tests systématiques sur l'évolution des données générées. La méthode proposée consiste dans un premier

3. Corpus codé selon la TEI, disponible sur le serveur Silfi de (`http://www.loria.fr/projets/Silfide/`)

temps à faire des tirages aléatoires d'énoncés et de construire automatiquement la grammaire LTAG associée à l'aide du système EGAL. Ceci permet d'étudier l'évolution de la taille de la grammaire LTAG (donnée par le nombre total de liens vers un schème) en fonction de l'augmentation d'énoncés pris en compte. Ainsi, plus on se rapproche d'une asymptote horizontale plus la couverture du sous-langage est bonne. La figure 3 donne l'évolution observée pour le corpus Gocad.

5.4. Résultats d'analyse

Nous présentons ici le résultat d'analyse sur le corpus de test et un exemple des statistiques qu'il est possible d'obtenir à l'aide de l'atelier, en fin de traitement, et après une mise au point de la grammaire (tables 3 et 4). En plus d'une comparaison des temps d'analyse, le système fournie une métrique concernant la robustesse des résultats. L'extension moyenne des îlots est la taille moyenne des chaînes reconnues pour l'ensemble des analyses partielles maximales [4]. Plus ce type d'îlot est étendu, plus complet sera l'arbre de dérivation associé.

Corpus	% analyse complètes	Nb moyen d' analyse/énoncé
Gocad	78.31	2.06

TAB. 3 – *Résultat global de l'analyse du corpus Gocad à l'aide de la grammaire LTAG extraite*

Algorithme	temps moyen d'analyse /énoncé (ms)	extension moyenne d'îlot
CKY prédictif	87	2.55
Analyse par connexité	58	2.79

TAB. 4 – *Comparaison entre deux algorithmes d'analyse ascendants*

6. Futures évolutions

L'objectif à moyen terme est de faire évoluer le système vers la prise en compte de la sémantique prédicative (syntaxe profonde) à l'aide de TAG synchrones (Shieber & Schabes, 1994). L'intégration des contraintes sémantiques dans le processus d'analyse sera permis par la synchronisation, et reposera sur le développement des deux points suivants :

- l'adaptation des algorithmes d'analyse pour la prise en compte des synchronisations dynamiques ;
- l'adaptation des outils de conception et de description en vue de définir la synchronisation des données statiques.

Il sera alors possible d'étendre la spécialisation des données à tout le niveau linguistique et d'étudier l'injection, au cours de ce processus, de contraintes liées à la sémantique de l'application.

4. Une analyse partielle maximale correspond, dans le cadre d'une analyse par *chart*, à un *item* qui n'est origine d'aucun autre *item.*

Références

ABEILLÉ A., DAILLE B. & HUSSON A. (1994). FTAG : An implemented Tree Adjoining grammar for parsing French sentences. In *TAG+3*, Paris.

BOD R. (1995). *Enriching Linguistics with Statistics : Performance Models of Natural Language*. PhD thesis, University of Amsterdam.

BRIFFAULT X., CHIBOUT K., SABAH G. & VAPILLOM J. (1997). An Object-Oriented Linguistic Engineering Environment using LFG and Conceptual Graphs. In *International Workshop ENVGRAM 97*, Madrid, Spain.

CANDITO M.-H. (1996). A principle-based hierarchical representation of LTAGs. In *COLING'96*, Copenhagen, Denmark.

CANDITO M.-H. (1999). *Structuration d'une grammaire LTAG : application au français et à l'italien*. PhD thesis, University of Paris 7.

CHAPELIER L., FAY-VARNIER C. & ROUSSANALY A. (1995). Modelling an Intelligent Help System from a Wizard of Oz Experiment. In *ESCA Workshop on Spoken Dialogue Systems*, Vigso, Danemark.

CIRAVEGNA F., LAVELLI A., PETRELLI D. & PIANESI F. (1997). Participatory Design for Linguistic Engineering: the case of the Geppetto Development Environment. In *Workshop ENVGRAM*, Madrid, Spain.

DEVILLE G. (1989). *Modelization of task-Oriented Utterances in a Man-Machine Dialogue System*. PhD thesis, University of Antwerpen, Belgique.

DORAN C., EGEDI D., HOCKEY B. A., SRINIVAS B. & ZAIDEL M. (1994). XTAG System - A Wide Coverage Grammar for English. In *COLING*, Kyoto, Japan.

EVANS R. & WEIR D. (1997). Automaton-based Parsing for Lexicalized Grammars. In *Fifth International Workshop on Parsing Technologies*, Cambridge, Mass.

LOPEZ P. (1998a). A LTAG grammar for parsing incomplete and oral utterances. In *European Conference on Artificial Intelligence (ECAI)*, Brighton, UK.

LOPEZ P. (1998b). Analyse guidée par la connexité de TAG lexicalisées. In *Conférence sur le Traitement Automatique du Langage Naturel (TALN'98)*, Paris, France.

PRICE P., MOORE R., MURVEIT H., PEREIRA F., BERNSTEIN J. & DALRYMPLE M. (1989). The integration of speech and natural language in interactive spoken language systems. In *Proceeding of Eurospeech*, Paris, France.

ROUSSEL D. & HALBER A. (1997). Filtering errors and repairing Linguistic Anomalies for Spoken Dialogue Systems. In *Workshop on Interactive Spoken Dialog Systems : ACL/EACL*, p. 74–81, Madrid.

SCHABES Y. (1992). Stochastic Lexicalized Tree Adjoining Grammars. In *COLING*, Nantes, France.

SCHABES Y. (1994). Left to Right Parsing of Lexicalized Tree Adjoining Grammars. *Computational Intelligence*, **10**, 506–524.

SHIEBER S. & SCHABES Y. (1994). Restricting the weak-generative capacity of synchronous tree-adjoining grammars. *Computational Intelligence*, **10**, 371–385.

SRINIVAS B. (1997). *Complexity of lexical descriptions and its relevance to partial parsing*. PhD thesis, University of Pennsylvania, Philadelphia.

VAN NOORD G. & BOUMA G. (1997). Hdrug. A flexible and Extendible Development Environment for Natural Language Processing. In *ENVGRAM Workshop*, Madrid, Spain.

VAN NOORD G., BOUMA G., KOELING R. & NEDERHOF M.-J. (1998). Robust Grammatical Analysis for Spoken Dialogue Systems. *Natural Language Engineering*, **1**, 1–48.

VIJAY-SHANKER K. & WEIR D. J. (1993). Parsing some constrained grammar formalisms. *Computational Linguistics*, **19**, 591–636.