# Analyse syntaxique combinant deux formalismes au sein d'un Chart hiérarchique

de Pierre-Olivier EL GUEDJ et Pierre NUGUES, Caen (F)

Membres du laboratoire GREYC, commun à l'Institut des Sciences de la Matière et du Rayonnement et à l'Université de Caen, Caen (F).

*elguedj@greyc.ismra.fr*
*pnugues@greyc.ismra.fr*

**Résumé**
Dans cet article, nous décrivons un analyseur combinant une grammaire syntagmatique, des règles de précédence et une grammaire de dépendances. Le système se fonde sur une structure de Chart divisée en plusieurs niveaux hiérarchiques. Ces niveaux correspondent à différents domaines d'analyse gérés par trois modules qui collaborent en apportant leur compétence linguistique propre. Le *module de segmentation syntagmatique* opère un découpage de la phrase pour donner un treillis composé des groupes syntaxiques (Sujet, verbe, compléments circonstanciels, etc.), des mots de structure (conjonction, …) et des signes de ponctuation. Le *module d'élagage* permet d'effectuer une sélection parmi les éléments du treillis afin de ne considérer que ceux qui présentent une cohérence avec le reste de la phrase. Enfin, le *module d'analyse des dépendances* relie les syntagmes afin de construire une représentation structurelle de la phrase.

**Summary**
In this paper, we describe a parser combining a phrase structure grammar, a set of precedence rules and a dependency grammar. The system is based on a Chart split in several hierarchical levels. These levels correspond to different stages of the parsing process managed by three modules. Each of these modules collaborate to the parsing bringing its linguistic competence to the system. The *Segmentation Module* carries out a segmentation of the sentence to give a lattice of the constituents (Subject, Verb, Adverbial Complements, etc.), structural words (conjunctions, …) and punctuation signs. The *Pruning Module* carries out a selection among the lattice elements in order to retain only the segments which are well integrated with the rest of the sentence. At last, the *Dependency Parser Module* links these segments to construct the dependency graph of the sentence.

## *1 Introduction*

Cet article décrit un analyseur syntaxique utilisant deux méthodes d'analyses complémentaires (El Guedj 1996; El Guedj & Nugues 1996; El Guedj & Nugues 1994). Lorsqu'on considère la langue française, on peut distinguer deux niveaux de structures syntaxiques. Le premier niveau correspond à des structures dont l'ordre des mots est très précisément défini : il s'agit de ce qu'on appelle communément les groupes syntaxiques (nominal, verbal, prépositionnel). Ces entités syntaxiques sont caractérisées par une certaine rigidité structurelle et l'utilisation d'une grammaire formelle (Chomsky, 1957) s'avère être un moyen adéquat permettant de les décrire puis les repérer au sein de la phrase.

Le second niveau de structure concerne les relations qui existent entre ces groupes syntaxiques. Du fait de la mobilité de certains de ces syntagmes au sein de la phrase, une grammaire de dépendances (Tesnière, 1959; Mel'cuk, 1988; Covington, 1990) se révèle être plus appropriée pour traiter ce second niveau structurel. Une telle grammaire permet d'établir les différentes relations hiérarchiques et fonctionnelles qui existent entre les divers groupes syntagmatiques.

L'utilisation concertée de ces deux paradigmes de représentation syntaxique permet d'optimiser la reconnaissance syntaxique des phrases et d'améliorer la robustesse des analyseurs. De plus, l'application d'une technique similaire à celle des étiqueteurs lexicaux (Chanod & Tapanainen, 1995) à un niveau non plus lexical mais syntaxique permet encore d'améliorer le processus d'analyse des phrases en éliminant les segments ne pouvant s'intégrer avec le reste de la phrase.

Nous avons évalué cet analyseur sur un corpus d'ordres oraux de navigation ainsi que sur une cinquantaine de phrases d'un corpus de comptes rendus médicaux provenant du Centre de traitement anti-cancéreux François Baclesse de Caen.

## *2 Architecture du système et principe de fonctionnement*

L'architecture du système d'analyse syntaxique se développe autour d'une structure de *Chart* (Kaplan, 1973; Kay, 1973; Gazdar & Mellish, 1989). Cette structure permet de stocker les hypothèses ainsi que les conclusions partielles de l'analyse en cours. Trois modules agissent directement sur les données contenues dans le Chart : *le module de segmentation syntagmatique, le module d'élagage et le module d'analyse des dépendances* (Figure 1).

Certains de ces modules construisent des hypothèses et les développent pour aboutir à des conclusions partielles. D'autres, au contraire, se chargent d'effectuer un tri parmi ces conclusions afin de favoriser celles qui sont admissibles et de mettre de côté celles présentant un défaut de cohérence ou de vraisemblance.

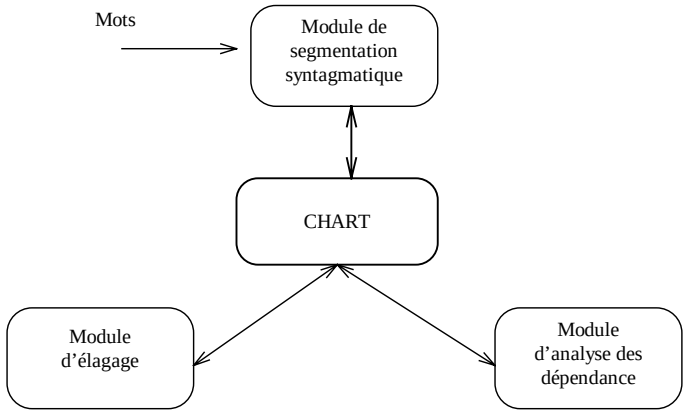

**Figure 1** Schéma conceptuel du système

### *3 L'organisation hiérarchique du Chart*

Du fait de l'interaction de plusieurs modules sur le Chart, nous avons divisé le Chart en plusieurs niveaux conceptuels à partir desquels les différents modules peuvent intervenir (Figure 2). Notre Chart organise les informations en fonction de la richesse de leur intérêt et du niveau de complexité qu'elles représentent.

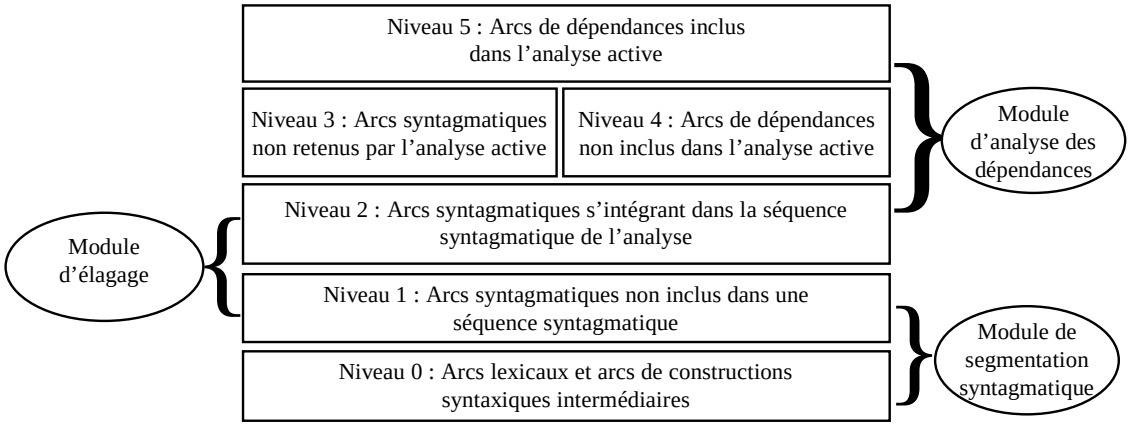


**Figure 2** Interaction des trois modules avec les différents niveaux hiérarchiques du Chart

Le niveau 0 est réservé aux arcs portant des hypothèses et des conclusions non syntagmatiques (Arcs lexicaux ou constructions syntaxiques intermédiaires). Le niveau 1 correspond au premier niveau syntagmatique. C'est à ce niveau que le module de segmentation syntagmatique dépose ses conclusions. Le module d'élagage examine l'ensemble de ces conclusions et déplace au niveau 2 les arcs syntagmatiques ayant satisfait aux conditions d'acceptabilité exigée par ce module. Les syntagmes ne remplissant pas les conditions requises sont conservés au niveau 1 qui constitue donc une zone d'archivage des conclusions non retenues pour la suite de la résolution de l'analyse.
Le niveau 5 est réservé aux structures de données intégrant les relations de dépendance et de rattachement mises en place par le module d'analyse des dépendances. Ces relations de dépendance concernent exclusivement les arcs syntagmatiques présents au niveau 2. Le module d'analyse des dépendances peut générer plusieurs analyses possibles de la phrase si elle présente des ambiguïtés au niveau de sa structure. Toutes ces analyses sont consultables mais on ne peut activer qu'une seule analyse à la fois. Elle peut correspondre à un choix établi par le système, l'utilisateur ou un module externe pour une étude ou un traitement particulier (affichage, analyse sémantique...). Les niveaux 3 et 4 servent respectivement à archiver les arcs syntagmatiques et les arcs de dépendance incompatibles ou non utilisés par l'analyse active.

### *4 Le module de segmentation syntagmatique*

Le *module de segmentation syntagmatique* repère les groupes syntaxiques présents dans la phrase. Il contient un analyseur syntagmatique dérivé des algorithmes utilisés habituellement dans les systèmes à base de Chart (Earley, 1970; Wirén, 1992). Il accomplit une analyse de gauches à droite. Lors de l'ajout d'un mot au Chart, l'algorithme classique d'Earley considère

toute la phrase déjà analysée pour tente de combiner l'arc du nouveau mot avec ceux déjà présents dans le Chart.
Notre algorithme de segmentation cherche à diminuer cet espace de recherche en ne concentrant l'analyse formelle que sur un segment de la phrase : le syntagme en cours d'analyse (Figure 3). Cette portion de phrase, appelée également *fenêtre d'analyse*, est délimitée par les deux nœuds (ou *vertex*) du Chart entourant les mots concernés. On nomme le nœud de gauche, le *point d'ancrage*. C'est à partir de lui que le module fonde son analyse formelle. L'analyse syntaxique s'effectuant de gauche à droite au fur et à mesure de la réception des mots, le nœud de droite correspond toujours au dernier nœud du Chart. Un nœud ne peut être un point d'ancrage qu'à deux conditions que nous appellerons les conditions d'ancrage :

- Soit le nœud considéré correspond au premier nœud du Chart;
- Soit il s'agit d'un nœud relié transitivement au nœud 0 par une composition d'arcs inactifs porteurs chacun d'un groupe syntagmatique complet.

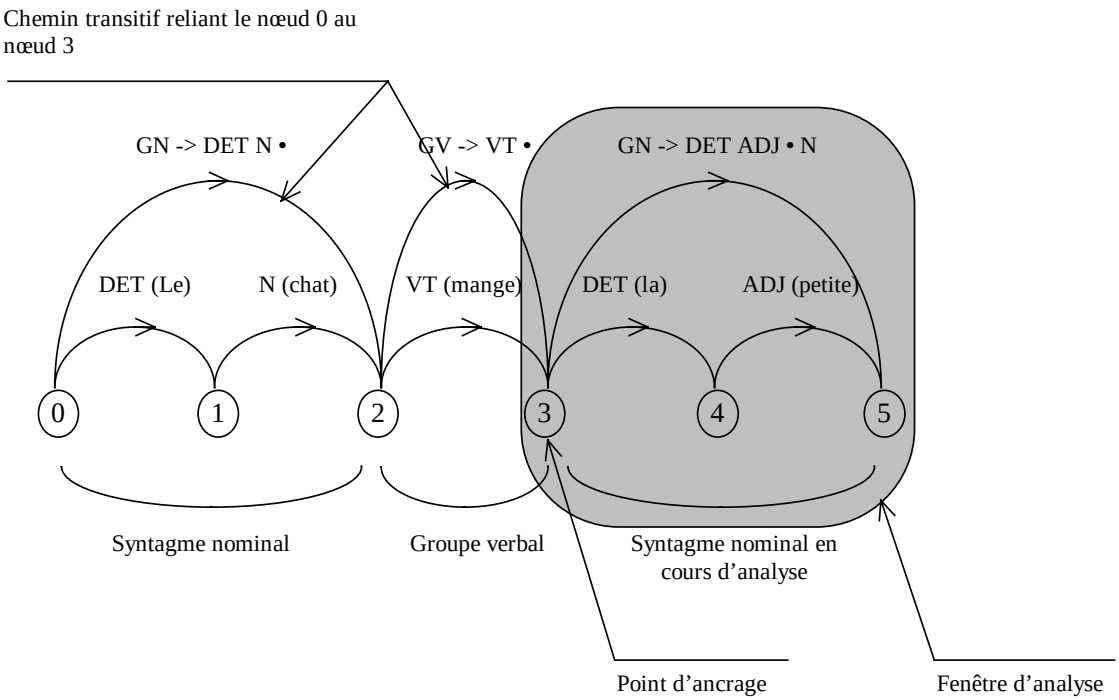


**Figure 3** Exemple de segmentation syntagmatique en cours de réalisation. GN décrit un groupe nominal et GV, un groupe verbal. DET, N, VT, et ADJ sont les abréviations respectives de déterminant, nom, verbe transitif et adjectif.

Au commencement de l'analyse, le point d'ancrage correspond au nœud 0. Au fur et à mesure que les mots sont analysés, le module construit le ou les groupes syntaxiques partant de ce point d'ancrage. Quand le dernier mot reçu ne peut s'intégrer dans le groupe syntagmatique en cours d'analyse, on considère alors que ce mot appartient à un autre syntagme. On déplace alors le point d'ancrage vers la droite sur un nœud constituant la fin d'un groupe

syntagmatique et on relance le processus. Si on ne trouve pas un tel nœud, on en déduit alors que ce dernier mot est incorrect et on le rejette. La segmentation syntagmatique s'effectue par bonds successifs vers les extrémités des syntagmes déjà analysés. La segmentation appliquée à la phrase ambiguë

*la belle ferme le voile*

permet d'obtenir un treillis des groupes syntagmatiques présents dans la phrase (Figure 3).

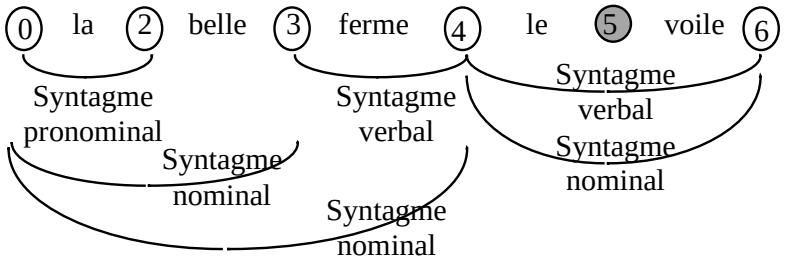


**Figure 4** Treillis final des syntagmes présents dans la phrase

### *5 Le module d'élagage*

Lorsqu'on considère le treillis de la Figure 4, on remarque au moins un syntagme dont la présence ne semble pas se justifier. En effet, l'interprétation du mot *la* en tant que syntagme pronominal ne peut être considérée comme acceptable pour deux raisons. En premier lieu, il n'existe pas de phrases affirmatives où un tel pronom est suivi par un adjectif; un tel syntagme est généralement placé entre le groupe sujet et le groupe verbal. De plus, même en considérant des formes syntaxiques ou stylistiques autorisant une telle construction, on remarque que ce syntagme s'incorpore mal à la phrase, car il n'est suivi par aucun autre syntagme. Il s'agit donc d'un résultat que l'on peut assurément mettre de côté du fait de son incompatibilité avec le reste de la phrase. Cette analyse de la cohérence d'une solution est réalisée par le *module d'élagage.*

Le travail effectué par ce module repose sur l'application d'un principe simple qui consiste à explorer le voisinage immédiat d'un syntagme pour vérifier s'il forme, avec ses voisins, une séquence acceptable. Cette technique s'apparente à celles utilisées dans les programmes d'étiquetage lexical (Brill, 1995), sauf que, dans notre cas, on se place à un niveau syntagmatique. Ce contrôle s'effectue par des règles de précédence qui indiquent les différents bigrammes syntagmatiques possibles (Figure 5).

Un syntagme n'est acceptable que si au moins l'un de ses prédécesseurs directs forme avec lui un bigramme syntagmatique correct. On applique aussi la réciproque de ce principe : un syntagme ne pouvant s'associer avec aucun de ses successeurs est considéré comme incorrect.

| Séquences admissibles | Explications | Exemples |
|---|---|---|
| GN GV | Un groupe verbal peut suivre un groupe nominal | ***Je vois*** *une voiture* |
| PRONOM_ANTE GV | Un pronom objet peut précéder un groupe verbal | *Je* ***la vois*** |
| GN PRONOM_ANTE | Un pronom objet antéposé peut suivre un groupe nominal | ***Je le*** *regarde* |

**Figure 5** Exemples de règles de précédence

On effectue donc un double balayage des séquences syntagmatiques pour que ce contrôle soit réalisé sur chacun des deux côtés de chaque syntagme. Les segments ne possédant pas de prédécesseurs ou de successeurs (à l'exception évidente des syntagmes de début et de fin de phrase) sont également écartés. À cause de ce manque de voisin, ils ne font manifestement pas partie d'une séquence syntagmatique reliant les deux extrémités de la phrase et ne peuvent donc s'intégrer à l'ensemble de celle-ci (Figure 6).

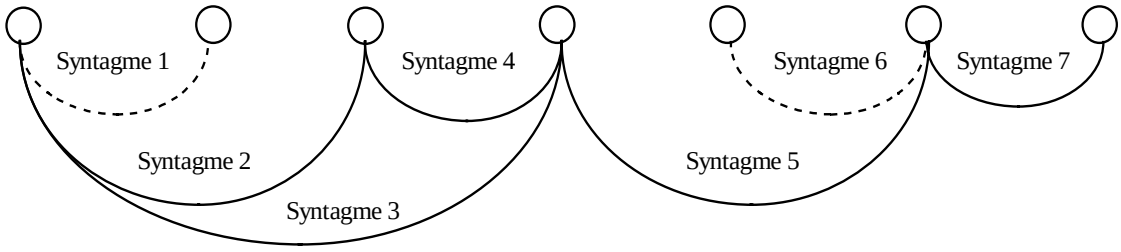


**Figure 6** Treillis comportant deux syntagmes non intégrables au reste de la phrase par manque de voisins : les syntagmes 1 et 6.

La mise à l'écart d'un ou plusieurs arcs syntagmatiques peut entraîner que leurs voisins ne soient plus acceptables. Le programme effectue donc un balayage des différents segments de gauche à droite puis de droite à gauche jusqu'à ce que l'on ne constate plus d'exclusion.

### *6 Le module d'analyse des dépendances*

Le *module d'analyse des dépendances* se charge d'accomplir la dernière étape de l'analyse à partir des éléments du niveau 2. Pour cela, il cherche à établir, au niveau 3 du Chart, des liaisons de dépendance entre les divers syntagmes retenus. À cause des diverses ambiguïtés qui peuvent persister à ce niveau de l'analyse, l'algorithme de dépendance se fonde sur un système de recherche en profondeur. À chaque étape de l'analyse, il choisit la dépendance ou le syntagme le plus probable. Ce choix repose sur l'application d'heuristiques topologiques. Quand un choix entre plusieurs possibilités se présente au cours de l'analyse, le système effectue un tri sur les différentes options qu'il peut suivre. Les heuristiques topologiques utilisées pour organiser ce tri sont les suivantes :

- Quand plusieurs syntagmes sont en concurrence au départ d'un même sommet, on privilégie le syntagme le plus long.
- Quand plusieurs dépendances sont en lice pour établir une tête – un régissant – sur un syntagme, on privilégie la dépendance pour laquelle la distance entre le syntagme à rattacher et sa tête est la plus courte.

L'application de ces heuristiques permet d'obtenir une liste ordonnée des différentes possibilités. Après cette phase d'ordonnancement, l'algorithme poursuit l'analyse en considérant l'hypothèse en tête de liste. En renouvelant ce processus sur tous les segments restants, on aboutit à une première analyse complète. On stocke alors la solution trouvée et on compte le nombre de syntagmes sans tête qu'elle contient. Ce nombre correspond au critère que nous avons retenu pour évaluer la qualité d'une analyse syntaxique : plus ce nombre est faible, meilleure est la fiabilité de l'analyse.

L'algorithme revient ensuite sur le dernier choix laissé en suspens et explore une nouvelle analyse en considérant l'hypothèse suivante. Si, au cours d'une telle exploration, on s'aperçoit que le nombre de syntagmes sans tête établis par ce parcours dépasse celui de la meilleure solution, on abandonne cette analyse. Si, par contre, une exploration débouche sur une solution équivalente voire meilleure que la précédente, on stocke cette nouvelle solution et on met à jour le score établi précédemment. De cette manière, on restreint l'espace de recherche en ne s'attachant qu'aux solutions jugées les meilleures à un instant donné de l'analyse. La Figure 6 présente une illustration du résultat généré par l'analyseur pour une des phrases du corpus.

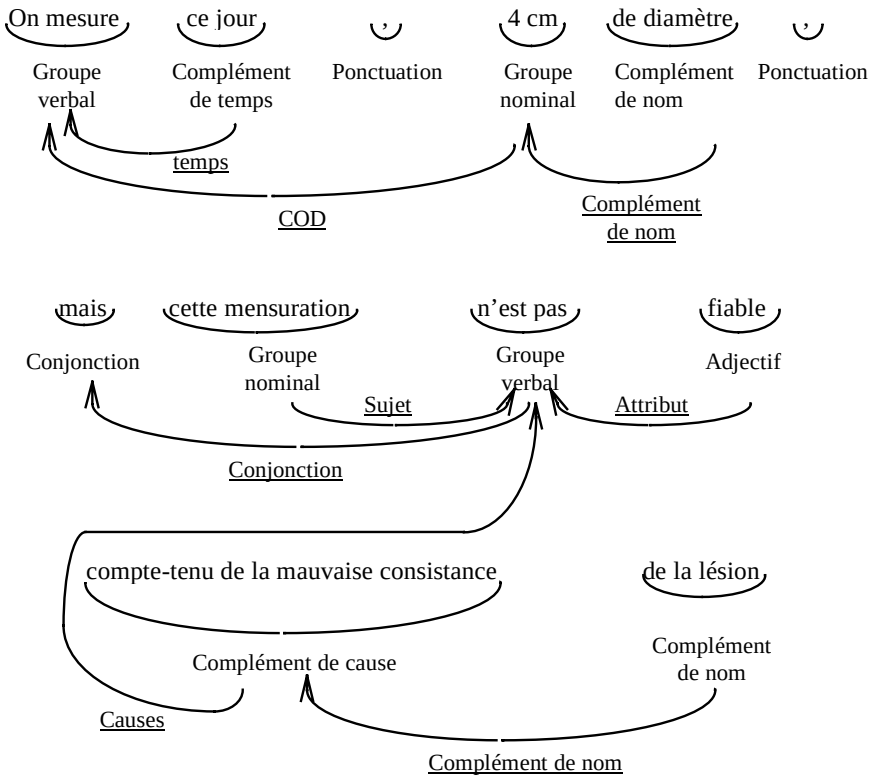


**Figure 7** Exemple de mise en relation des segments pour une des phrases du corpus

### *7 Implantation, résultats et applications*

Le système décrit a été implanté en C++ en tirant parti de la programmation orientée objet. Il a été développé avec la perspective d'une utilisation sous divers environnements parmi lesquelles Windows et Unix. La version sous Windows bénéficie d'une interface graphique permettant d'étudier et d'améliorer dans de bonnes conditions le fonctionnement du programme.

L'analyseur syntagmatique présent dans le module de segmentation a été intégré au sein d'un système de dialogue oral. Cette application consiste en la mise en œuvre d'un agent conversationnel de navigation permettant à un utilisateur de se déplacer à la voix au sein d'un environnement de réalité virtuelle (Godéreaux et al., 1996). L'analyseur réalise une analyse purement formelle permettant de couvrir la quasi totalité des ordres de déplacement du corpus que nous avons collecté pour cette application.

Nous avons également évalué notre système sur un échantillon de 54 phrases issues d'un corpus de comptes rendus médicaux. La combinaison des deux premiers modules a donné de très bons résultats. Le module de segmentation a détecté plus de 93% des segments attendus. Une seule phrase de l'échantillon n'a pu être segmentée totalement par l'analyseur à cause d'une incise entre une préposition et son groupe nominal. Sans ce cas particulier, le taux de couverture des segments correctement détectés aurait atteint plus de 99,5%.
Ces résultats sont cependant accompagnés d'une surproduction de segments importante avec plus de 191% de segments incorrects par rapport au nombre de segments attendus. Le module d'élagage apporte un gain incontestable au niveau de la précision du système en éliminant plus de 82% de ces segments incorrects, faisant ainsi tomber le taux de surproduction aux alentours de 33%. Les résultats obtenus par le module de dépendance portent sur les phrases ayant passé avec succès les deux premières phases de l'analyse. Nous avons donc exclus de l'échantillon la phrase qui avait posé quelques difficultés au module de segmentation à cause d'une incise rejetée par la grammaire de l'analyseur. Certaines des 53 phrases restantes pouvant être interprétées de multiples manières, un ensemble de 107 analyses plausibles d'un point de vue sémantique a été établi comme modèle d'analyse de l'échantillon. Sur ces 107 solutions, l'analyseur en a trouvé 88 sans aucune erreur (82,2 % ). Si on fait abstraction des erreurs mineures de catégorisation ou de typage des segments ou des relations, ce taux de réussite monte à 83,2 %. Enfin, si on ajoute également les solutions trouvées de manière partielle par l'analyseur sans erreurs majeures au niveau de la segmentation et de la mise en relation, le pourcentage monte à 92,5 %.

### *8 Conclusion*

Nous avons présenté un analyseur syntaxique fondé sur les techniques de Chart. L'analyse proprement dite se décompose en trois phases. Grâce à ces trois phases de traitement, le système peut construire une analyse progressive. Celle-ci débute au niveau lexical, se poursuit au niveau syntagmatique et se termine au niveau de l'organisation structurelle de la phrase elle-même. Cette décomposition permet d'accomplir une analyse plus rapide et robuste.
L'organisation à plusieurs niveaux du Chart favorise l'analyse partielle et devrait permettre la mise en œuvre d'algorithmes de reprise sur erreur pour traiter les phrases agrammaticales. Enfin, nous avons confronté un sous-ensemble de notre système à la navigation orale dans les mondes virtuels ainsi qu'à un échantillon de phrases issues d'un corpus de comptes rendus médicaux.
Les résultats que nous avons obtenus nous permettent de valider notre démarche et d'envisager une analyse complète ou partielle de rapports médicaux sans restriction. Une telle analyse devrait permettre notamment la mise en œuvre d'outils perfectionnés de recherche d'informations et de compréhension automatique de textes.

### *Références*

Brill E. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging, *Computational Linguistics*, 21(4):543-565, 1995.

Chanod, J.P. & Tapanainen, P., Tagging French – Comparing a Statistical and Constrainst-Based Method, *Proceeding of the Seventh Conference of the European Chapter of the ACL (EACL'95)*, page 149-156, Association for Computational Linguistics, Dublin, 1995.

Chomsky, N. *Syntactic Structures*, La Haye, Mouton, 1957.

Covington, M. Parsing discontinuous constituents in dependency grammar, *Computational Linguistics* 16:234-236, 1990.

Earley, J. An efficient context-free parsing algorithm. *Communication of the ACM* 13:94-102, 1970.

El Guedj, P.O. & Nugues, P., A Chart parser to analyze large medical corpora, In: *Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Baltimore, pp. 1404-1405, November 1994.

El Guedj P.-O., *Analyse syntaxique par chart combinant règles de dépendance et règles syntagmatiques*, Mémoire de thèse, Université de Caen, 10 septembre 1996.

El Guedj P.-O. & P. Nugues, Un analyseur syntaxique combinant plusieurs formalismes au sein d'un chart hiérarchique, *Actes du Colloque RÉCITAL '96*, pp. 83-91, Dourdan, septembre 1996.

Gazdar, G. & Mellish, C. *Natural language Processing in Prolog: an introduction to computational linguistics.* Wokingham: Addison-Wesley, 1989.

Godéreaux, C., Diebel, K., El-Guedj, P.O., Revolta, F. & Nugues, P.: An Interactive Spoken Dialog Interface to Virtual Worlds, in: *Linguistic Concepts and Methods in CSCW*, J.H. Connolly & L. Pemberton eds, Chapter 13, pp. 177-200, Springer Verlag, 1996.

Kaplan, R.M., A general syntactic processor, *In natural Language Processing*, pp. 193-241, Randall Rustin (ed.), Algorithmics Press, New York, 1973.

Kay, M., (1973) The MIND system, *In Natural Language Processing,* pp. 155-188, Randall Rustin (ed.), Algorithmics Press, New York, 1973.

Mel'cuk I. A.: *Dependency Syntax: Theory and Practice*, State University of New York Press, 1988.

Tesnière, L., *Éléments de la syntaxe structurale*, Paris, Klincksieck, 1959.

Wirén, M., *Studies in Incremental Natural-Language Analysis*, Dissertation N°. 292, Linköping Studies in Science and Technology, University of Linköping, Suède, 1992.