

# **MULTI-ANALYSE**

## **vers une analyse syntaxique plus fiable**

Laura Monceaux et Anne Vilnat

LIMSI/CNRS – Université Paris Sud  
Bat 508 – BP 133 91403 Orsay Cedex  
{monceaux, vilnat}@limsi.fr

### **Résumé – Abstract**

Dans cet article, nous proposons de montrer que la combinaison de plusieurs analyses syntaxiques permet d'extraire l'analyse la plus fiable pour une phrase donnée. De plus, chaque information syntaxique sera affectée d'un score de confiance déterminé selon le nombre d'analyseurs syntaxiques la confirmant. Nous verrons que cette approche implique l'étude des différents analyseurs syntaxiques existants ainsi que leur évaluation.

In this paper, we propose an algorithm of combination between several syntactic parses in order to extract the most reliable syntactic parse for the sentence given and to affect these syntactical informations with the confidence rate, determined with regard to the number of parsers returning the same information. This approach involves the study of syntactic parsers and their evaluation.

### **Mots Clés - Keywords**

Analyseurs syntaxiques, Combinaison d'informations, Evaluation  
Syntactic parsers, Information combination, Evaluation

## **1 Introduction**

Dans le cadre du développement d'un système de question – réponse dans le groupe LIR (Ferret et al, 2002), nous nous sommes particulièrement intéressées aux analyseurs syntaxiques dits « robustes » dans le but de déterminer l'analyseur syntaxique le plus approprié pour ce type d'application. En effet, l'analyse de la question dans le système QALC dépend de connaissances syntaxiques et sémantiques (Monceaux et Robba, 2002). Mais aujourd'hui, les analyseurs syntaxiques ne sont pas nécessairement tous fiables pour analyser totalement ce type de questions. Or, la fiabilité des informations syntaxiques utilisées dans le module d'analyse des questions est essentielle ; plus celles-ci seront fiables, plus les informations extraites de

l'analyse de la question le seront. Même si l'on dispose d'un protocole d'évaluation des analyseurs syntaxiques permettant de détecter l'analyseur le plus fiable pour notre type d'applications, cela ne garantira pas que toutes les informations retournées par celui-ci seront fiables. Ainsi nous avons eu l'idée, afin d'améliorer les résultats de l'analyse des questions, d'utiliser une multi-analyse permettant d'obtenir l'analyse la plus fiable pour une phrase donnée et d'affecter un score de confiance à chaque information retournée.

Cette approche de combinaison entre plusieurs informations de même type a déjà connu de nombreux succès notamment dans la combinaison de plusieurs transcriptions de la parole (Fiscus, 1997), (Schwenk et Gauvain, 2000) grâce à une stratégie de vote simple ou encore la combinaison d'étiquetages morpho-syntaxiques (Marquez et Padro, 1998), (Paroubek and Rajman, 2000) permettant d'obtenir un étiquetage plus fiable mais également de fournir des corpus annotés. En ce qui concerne la combinaison d'analyseurs syntaxiques, une telle approche fait actuellement défaut. En effet, cette combinaison semble plus difficile car les analyseurs syntaxiques peuvent retourner différents types de réponse : une segmentation en constituants ou/et un ensemble de relations syntaxiques entre mots voire entre groupes de mots. Dans un premier temps, notre module d'analyse des questions utilisant essentiellement le découpage en constituants, nous avons développé une multi-analyse dans le but de reconnaître les groupes verbaux, nominaux et prépositionnels les plus fiables. Toutes les informations syntaxiques obtenues par la combinaison de plusieurs analyses seront obtenues par un consensus entre plusieurs analyseurs syntaxiques.

Après avoir détaillé quels analyseurs étaient appropriés pour une telle combinaison, nous présenterons la stratégie de combinaison que nous avons mise au point et nous illustrerons cette stratégie sur un exemple. La présentation des résultats obtenus pour l'évaluation des résultats obtenus par combinaison sera suivie d'une discussion sur les perspectives de ce travail.

## **2 Quels analyseurs syntaxiques utiliser pour la combinaison ?**

Dans un premier temps, nous avons étudié les différents analyseurs existants pour détecter les analyseurs les plus appropriés pour notre type d'application. En effet, dans le projet MULTITAG (Paroubek and Rajman, 2000), il a été remarqué que l'utilisation de systèmes ayant à peu près les mêmes compétences était préférable pour réaliser le meilleur multi-étiquetage pour une phrase donnée.

Pour réaliser notre première combinaison, nous avons été confrontées au choix des analyseurs à utiliser. Comme nous l'avons présenté plus haut, un analyseur robuste semble plus approprié pour une telle application, mais il faut également que l'analyseur retourne une segmentation en constituants. Le choix de l'analyseur pour l'anglais repose sur le fait que le système de question – réponse développé dans notre équipe de recherche (groupe LIR : Langages, Information et Représentations) est actuellement en anglais. Mais l'approche proposée sera applicable également pour le français. Ces analyseurs ont été choisis car chacun d'eux retourne une segmentation en constituants, information syntaxique sur laquelle nous voulons réaliser notre combinaison. Nous avons donc sélectionné quatre analyseurs syntaxiques robustes pour l'anglais :

- L'analyseur IFSP de Xerox (Aït-Mokhtar and Chanod, 1997)

- L'analyseur statistique Apple Pie (Sekine and Grishman, 1995)
- L'analyseur SCOL (Abney, 1996)
- L'analyseur IPS (Wehrli, 1992)

Cependant au regard des sorties obtenues par ces différents analyseurs, on remarque qu'il n'y a aucun accord sur la représentation des sorties d'une analyse syntaxique (voir Fig. 1.). La première étape pour faciliter la combinaison consiste à trouver un langage pivot qui va nous permettre de projeter chacune de ces sorties dans un même formalisme.

Analyseurs	Résultats
IFSP <sup>1</sup>	[SC [NP Which museum NP] [PP in Florence PP] :v was damaged SC] [PP by a major bomb PP] ?
APPLE-PIE <sup>2</sup>	(S (NP (NPL Which museum) (PP in (NLP Florence))) (VP was (VP damaged (PP by (NP (NPL a major bomb)))))) ?
SCOL <sup>3</sup>	[rc [orc0 [wdt which] [nx [nn museum]] [pp [in in] [np Florence]] [vx [bedz was] [vbn damaged]]] [pp [by by] [nx [dt-a a] [jj major] [nn bomb]]]] ?
IPS <sup>4</sup>	[CP [DP which [NP museum [PP in [DP Florence]]]] [C [TP [DP e] [T was [VP [FP [DP e] [F [VP damaged [PP by [DP a [NP [AJ major] [N bomb]]]]]]]]]]]] ?

Figure 1. Diversité des sorties des différents analyseurs syntaxiques

Le formalisme que nous avons décidé d'utiliser est le formalisme d'annotation décrit dans le projet PEAS du LIMSI dans lequel nous travaillons (Gendner and al, 2002). Le projet PEAS consiste à définir un protocole d'évaluation des analyseurs syntaxiques reposant sur la comparaison des sorties d'un analyseur avec un corpus annoté de référence. Un formalisme

<sup>1</sup> SC (Sentence Clause), NP (Noun Phrase), PP (Prepositional Phrase), :v (verbe) ...

<sup>2</sup> S (Sentence), NP (Noun Phrase), NLP (Proper Name), VP (Verbal Phrase), PP (Prepositional Phrase)...

<sup>3</sup> pp (prepositional phrase), nx (noun phrase), vx (verbal phrase), np (proper noun) ...

<sup>4</sup> DP (Determiner Phrase), VP (Verbal Phrase), PP (Prepositional Phrase), CP (Complementizer Phrase), TP (Tense Phrase)

d'annotation a été établi. La segmentation en constituants repose sur 6 groupes syntaxiques terminaux :

- Les noyaux verbaux (NV),
- Les groupes nominaux (GN),
- Les groupes prépositionnels (GP),
- Les groupes adjectivaux (GA),
- Les groupes adverbiaux (GR),
- Les groupes verbaux prépositionnels (PV)

Pour faciliter la combinaison, nous avons donc élaboré pour chaque analyseur syntaxique un algorithme de projection de leur segmentation dans le formalisme PEAS (développé en JAVA). L'élaboration de ces algorithmes de projection a entraîné une étude approfondie des différentes théories linguistiques employées par les analyseurs et a débouché sur l'élaboration d'un ensemble plus ou moins complexe de règles de réécriture ordonnées pour chaque analyseur (Monceaux, 2002). On peut notamment constater dans les règles de réécriture de l'analyseur IFSP que les informations pour la projection sont autant dans la segmentation proposée que dans les relations syntaxiques.

Résultats de l'analyseur IFSP	Résultats après réécriture
[PP liste-de-mots PP]	<GP> liste-de-mots </GP>
: v auxiliaire SC] [SC [NP pronom NP] : v verbe SC] && SUBJ (pronom auxiliaire)	<NV> auxiliaire pronom verbe </NV>
[NP pronom NP] :v verbe SC] && SUBJ (pronom, verbe)	<NV> pronom verbe </NV>
[NP liste-de-mots NP]	<GN> liste-de-mots </GN>

Figure 2 : Quelques règles de réécriture pour l'analyseur IFSP

Pour l'exemple de la figure 1, l'application des différentes règles de réécriture permettra de construire l'analyse syntaxique suivante à partir de l'analyse retournée par IFSP :

<GN> Which museum </GN> <GP> in Florence </GP> <NV> was </NV>  
<NV> damaged </NV> <GP> by a major bomb explosion </GP> ?

Ces algorithmes de projections vont non seulement nous permettre de projeter les résultats de chaque analyseur dans le formalisme PEAS dans le but de combiner les différents résultats pour une analyse syntaxique la plus probable, mais également nous servir pour évaluer les résultats obtenus par la combinaison par rapport à chaque analyseur syntaxique. Ces différentes

évaluations nous permettront ainsi de conclure si l'algorithme de combinaison nous permet d'obtenir de meilleurs résultats qu'un analyseur seul.

### 3 Combinaison entre plusieurs analyses syntaxiques

Dans cette section, nous allons expliquer plus précisément l'algorithme de combinaison que nous avons utilisé basé sur la notion de vote à la majorité : plus une information sera retournée par un grand nombre d'analyseurs syntaxiques, plus celle-ci sera fiable et donc affectée d'un fort taux de confiance. Successivement, l'algorithme de combinaison va essayer de déterminer les groupes verbaux les plus plausibles, les groupes prépositionnels et pour finir les groupes nominaux. En effet, grâce aux évaluations effectuées sur les différents analyseurs que nous présenterons dans la suite, nous avons remarqué que les noyaux verbaux étaient généralement mieux reconnus que les groupes prépositionnels, et ces derniers mieux que les groupes nominaux. Pour chaque information syntaxique recherchée, on réalisera donc un ensemble de phases.

#### 3.1 Construction de l'ensemble des groupes syntaxiques possibles

La première phase de cet algorithme consiste à sélectionner les analyseurs syntaxiques sur lesquels la combinaison va être réalisée. Puis pour chaque phrase, les résultats obtenus par chaque analyseur sont projetés dans le formalisme commun PEAS. L'algorithme de combinaison construit, à partir de ces différentes sorties, l'ensemble des groupes syntaxiques susceptibles d'être fiable. Cet ensemble est organisé par type syntaxique où chaque groupe syntaxique proposé sera accompagné du nombre d'analyseurs l'ayant reconnu. De plus, ces différents groupes seront également étiquetés d'une étoile s'ils ont été retournés par l'analyseur le plus approprié pour détecter ce type d'information suite à l'évaluation réalisée au préalable (que nous présenterons dans la suite).

A partir des résultats obtenus par les quatre analyseurs sélectionnés pour la question « What grape variety is used in Château Petrus Bordeaux ? », nous obtiendrons ainsi l'ensemble suivant :

Noyaux Verbaux	Groupes Prépositionnels	Groupes Nominaux
is – 4*	In Château Petrus Bordeaux – 2*	what grape variety – 2*
used – 4*	in Château – 2	Petrus – 2
		Bordeaux – 2

Figure 3 : Ensemble des groupes syntaxiques possibles pour la combinaison

Les groupes syntaxiques retournés par un seul analyseur sur les quatre seront automatiquement supprimés puisque considérés comme non fiables donc les groupes nominaux « what grape », « variety », « grape variety » n'ont pas été sélectionnés.

A partir de cet ensemble des groupes syntaxiques possibles, on va tenter de détecter les groupes syntaxiques les plus fiables et comme nous l'avons dit plus haut les noyaux verbaux dans un premier temps, les groupes prépositionnels puis les groupes nominaux.

### 3.2 Sélection des groupes syntaxiques les plus fiables

La sélection des groupes syntaxiques les plus fiables parmi l'ensemble des groupes syntaxiques possibles se réalise en deux phases :

1. Sélection des groupes syntaxiques les plus fiables : il s'agit ici de sélectionner les informations considérées comme les plus fiables, c'est-à-dire données par plus de la moitié des analyseurs. Dans notre exemple, il s'agira des groupes syntaxiques retournés par au moins trois des analyseurs. Si l'information syntaxique sélectionnée est retournée par le maximum des analyseurs, on lui attribuera un taux de confiance égal à 1 et de 0.75 si elle est retournée par seulement trois analyseurs.
2. Sélection des groupes syntaxiques dits « litigieux » : certains groupes retournés par seulement la moitié des analyseurs sont également sélectionnés mais avec un taux de confiance moindre. Si le groupe syntaxique étudié n'est pas en conflit avec un autre groupe de même type, celui-ci sera sélectionné et affecté d'un taux de confiance 0.5. Sinon le groupe syntaxique étudié est en conflit (un mot ou plusieurs mots sont présents dans plusieurs groupes de même type et retournés par la moitié des analyseurs), alors le groupe syntaxique retourné par l'analyseur le plus approprié pour détecter ce type d'information (marqué d'une étoile) sera sélectionné avec un taux de confiance 0.25.

A chaque fois qu'un groupe syntaxique est sélectionné, tout autre groupe possédant un ou plusieurs mots de ce groupe devra être définitivement supprimé de l'ensemble des groupes syntaxiques possibles. En effet, dans le formalisme PEAS, les groupes syntaxiques sont terminaux : donc quand un mot est associé à un groupe, il ne peut pas être affecté à un autre groupe.

### 3.3 Exemple de combinaison

La première étape de l'algorithme de combinaison consiste à déterminer l'ensemble des groupes syntaxique susceptibles d'être les plus fiables. Nous allons détailler les différentes étapes de sélection des groupes à partir de l'ensemble des groupes syntaxiques construits pour l'exemple « What grape variety is used in Château Petrus Bordeaux ? » (voir Fig. 3).

Comme nous l'avons vu, la première étape de l'algorithme consiste à détecter les groupes verbaux les plus fiables. Dans notre exemple, les deux groupes « is » et « used » sont retournés par la majorité des analyseurs, ils sont donc sélectionnés et affectés d'un poids de confiance 1.

L'étape suivante doit déterminer les groupes prépositionnels les plus fiables. Ici on se retrouve confronter à deux groupes litigieux contradictoires. En effet, « in Château Petrus Bordeaux » et « in Château » possèdent des mots en commun. Dans ce cas, le groupe syntaxique retourné

par l'analyseur le plus approprié pour détecter ce type d'information (marqué d'une étoile) sera sélectionné et affecté d'un taux de confiance 0.25.

Suite à ces deux premières étapes, on obtient l'ensemble des groupes syntaxiques présentés dans la figure 4. La sélection du groupe « in Château Petrus Bordeaux » comme groupe prépositionnel a entraîné la suppression des deux groupes nominaux « Petrus » et « Bordeaux » puisque les mots qu'ils comportent ont été affectés à un autre groupe syntaxique.

Noyaux Verbaux	Groupes Prépositionnels	Groupes Nominaux
<b>is – Conf : 1</b>	<b>in Château Petrus Bordeaux – Conf : 0.25</b>	What grape variety – 2*
<b>used – Conf : 1</b>	<del>in Château – 2</del>	<del>Petrus – 2</del>
		<del>Bordeaux – 2</del>

Figure 4 : Ensemble des groupes syntaxiques possibles après les deux premières étapes

La dernière étape réside dans le choix des groupes nominaux les plus fiables. Ici seul « What grape variety » peut être un groupe nominal. Comme il est retourné par la moitié des analyseurs et aucunement en contradiction avec un groupe du même type, il est sélectionné avec un taux de confiance de 0.5.

Finalement, pour la phrase « What grape variety is used in Château Petrus Bordeaux ? », l'algorithme de combinaison retournera l'analyse suivante :

*<GN Conf : 0.5> What grape variety </GN> <NV Conf : 1> is </NV> <NV Conf : 1> used </NV> <GP Conf : 0.25> in Château Petrus Bordeaux </GP> ?*

Cet exemple nous montre qu'à partir de quatre analyses (pas nécessairement correctes), on peut obtenir une analyse pertinente et donc fiable. Dans le but d'évaluer cet algorithme, nous avons réalisé une évaluation quantitative de la pertinence des résultats obtenus par la combinaison.

## 4 Évaluation de la combinaison

Pour évaluer les résultats obtenus par l'algorithme de combinaison, nous avons utilisé les bases du protocole d'évaluation des analyseurs syntaxiques du projet PEAS. Pour chaque type de groupe syntaxique, nous allons donc calculer les mesures de rappel et de précision. La mesure de rappel va nous permettre de conclure quant à la quantité d'informations correctes trouvées alors que la mesure de précision va nous permettre de valider la fiabilité des informations retournées.

Dans le protocole PEAS, chaque analyse est évaluée par rapport à une analyse annotée référence. Pour réaliser une première évaluation de l'algorithme de combinaison, nous avons donc annoté un petit corpus de 66 questions, plus précisément les noyaux verbaux, les groupes prépositionnels et les groupes nominaux. Ce corpus a été extrait de la campagne d'évaluation

TREC 2001 pour la tâche Question – Réponse en contexte : il regroupe 49 groupes nominaux, 77 noyaux verbaux et 38 groupes prépositionnels.

Dans un premier temps, nous avons étudié la fiabilité des résultats obtenus par chaque analyseur et par l’algorithme de combinaison grâce à la mesure de précision (voir Fig. 5). Cette mesure va non seulement nous permettre de voir si l’algorithme de combinaison permet d’avoir des informations plus fiables mais également de déterminer l’analyseur le plus approprié pour chaque type d’information. On constate qu’effectivement la combinaison des différentes analyses permet d’améliorer la fiabilité des résultats obtenus notamment en ce qui concernent les groupes nominaux.

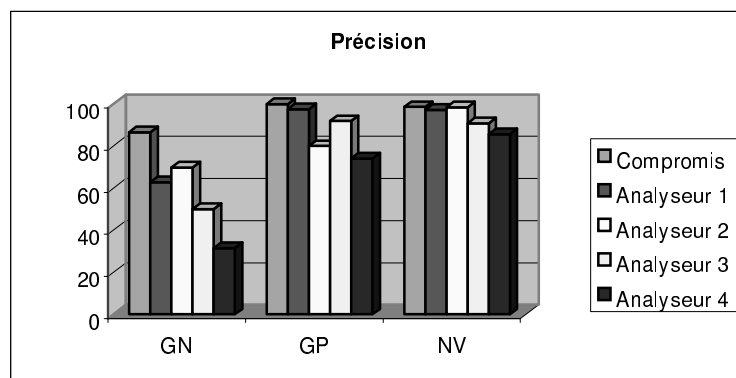


Figure 5 : Mesure de précision

Cependant si l’on voit que la fiabilité des informations a augmenté, il faut vérifier que l’algorithme nous permet d’avoir autant d’informations qu’un analyseur seul grâce à la mesure de rappel. Au vu des résultats obtenus (voir Fig. 6), on peut dire que les résultats sont satisfaisants : on a une bonne moyenne d’informations retournées par rapport aux résultats des analyseurs seuls.

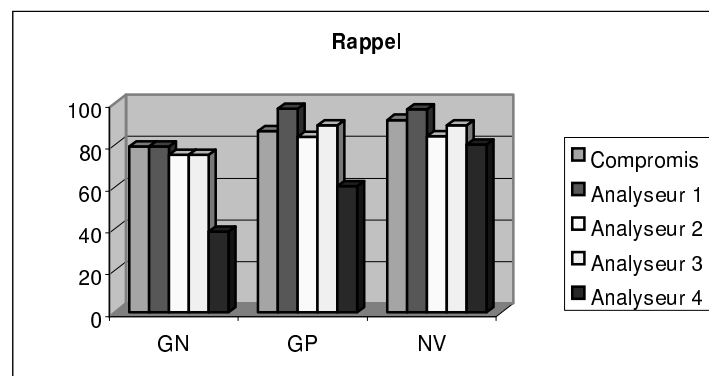


Figure 6 : Mesure de rappel

Les différentes évaluations réalisées nous permettent de conclure que sur ce corpus, l’algorithme de combinaison permet de retourner plus d’informations syntaxiques fiables qu’un analyseur seul.



Si l'on regarde de plus près les résultats obtenus par la combinaison par rapport aux résultats obtenus par l'analyseur 1 (celui ayant les meilleurs résultats lors de l'évaluation), on peut constater que l'algorithme de combinaison a permis de rattraper certaines erreurs réalisées par l'analyseur 1. Par exemple, pour la phrase « Name a company that flies. », le mot « name » est étiqueté comme groupe nominal par l'analyseur 1, alors que dans le compromis, celui-ci est étiqueté noyau verbal avec un taux de confiance de 0.75 :

*<NV Conf: 0.75> Name </NV> <GN Conf: 1> a company </GN> that  
<NV Conf: 1> flies </NV>*

Cependant, certaines erreurs ne peuvent être corrigées surtout si elles sont commises par une majorité d'analyseurs. Par exemple pour la question « How much did the future cost for the vintage 1989 ? », trois analyseurs retournent le groupe nominal « the future cost ». Donc il est impossible de corriger puisque plus de la moitié des analyseurs font cette erreur. L'algorithme de combinaison sélectionnera donc ce groupe avec un taux de confiance assez important :

*How much <NV Conf: 1> did </NV> <GN Conf: 0.75> the future cost </GN>  
<GP Conf: 0.75> for the vintage 1989 </GP> ?*

Cette première évaluation nous laisse espérer que la fiabilité d'une telle combinaison augmentera d'autant plus avec un corpus plus large.

## **5 Conclusion**

La combinaison d'informations de même type permet généralement d'augmenter l'efficacité des systèmes utilisés. Face à ce constat, nous avons donc développé un outil de combinaison entre plusieurs analyses syntaxiques et avons montré que l'évaluation réalisée sur les résultats obtenus par une combinaison entre quatre analyseurs syntaxiques sur un corpus de question permettait de valider cette hypothèse de départ. Cette combinaison permet effectivement d'augmenter la performance de l'analyse syntaxique retournée et plus particulièrement la segmentation en constituants. On peut espérer que sur un corpus plus large, ces résultats seront confirmés. La principale difficulté de cette proposition réside dans la projection des résultats de chaque analyseur dans un même formalisme, qui requiert une bonne connaissance de la théorie linguistique de chaque analyseur. Un des avantages de cette combinaison réside également dans le fait que les informations syntaxiques retournées sont accompagnées d'un taux de confiance, cela permettant aux utilisateurs de choisir les informations les plus pertinentes pour son application selon cette estimation de fiabilité. Notre motivation était partie en effet du constat que pour obtenir de bons résultats dans l'analyse des questions d'un système de question – réponse, les informations syntaxiques de la question devaient être fiables.

A moyen terme, nous avons l'intention de développer l'algorithme de combinaison notamment sur les types d'informations à combiner comme les relations syntaxiques entre mots ou / et groupes de mots. Mais nous voulons également tester notre algorithme sur un corpus plus large et plus divers. Cette évaluation nous permettra non seulement de confirmer les résultats obtenus par la première évaluation mais également nous permettra de voir si les analyseurs syntaxiques choisis sont de même performance. En effet, comme nous l'avons dit, il est plus intéressant de réaliser une combinaison sur des systèmes ayant des performances proches.

Nous verrons également si le nombre d'analyseurs utilisés est suffisant ou si l'augmentation du nombre des analyseurs permet d'augmenter la fiabilité des résultats. A plus long terme, cette proposition de multi-analyse pourra également servir de base au développement d'un corpus de référence pour l'évaluation des analyseurs syntaxiques.

## Références

- Abney S. (1996) Partial Parsing via finite-state cascades. *J. of Natural Language Engineering*, 2(4), 337-344.
- Aït-Mokhtar S., Chanod J.P. (1997) *Incremental finite-state parsing*. Actes ANLP-97, Washington.
- Gendner V., Illouz G., Jardino M., Monceaux L., Paroubek P., Robba I., Vilnat A. (2002) *A protocol for evaluating analyzers of syntax (peas)*. Proceedings of LREC-2002, Las Palmas, Canary.
- O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, I. Robba, A. Vilnat – *Recherche de la réponse fondée sur la reconnaissance du focus de la question*, Actes de TALN'2002, Nancy, France, 24-27 juin 2002, pages 307-316
- Fiscus J.G (1997) *A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)*. Actes IEEE Workshop on Automatic Speech Recognition and Understanding.
- Marquez and Padro (1998) *On the evaluation and comparison of taggers: the effect of noise in test corpora*. Actes COLING/ACL'98, Montreal, Canada.
- Monceaux L. (2002) *Adaptation du niveau d'analyse des interventions dans un dialogue – application à un système question – réponse*, Thèse de doctorat, Université Paris XI.
- Monceaux L., Robba I. (2002) *Les analyseurs syntaxiques : atouts pour une analyse des questions dans un système de question-réponse?* Actes TALN-2002, Nancy, France.
- Paroubek P., Rajman M. (2000) *Multitag: une ressource linguistique produit du paradigme d'évaluation*, Actes TALN-2000, Lausanne, Suisse.
- Sekine S., Grishman R. (1995) *A corpus-based probabilistic grammar with only two Non-terminals*. Actes IWPT-95, Prague.
- Schwenk H., Gauvain J.L. (2000) *Combining multiple speech recognizers using voting and language model information*. Actes ICSLP-2000, pages 915-918, Pekin.
- Wehrli E. (1992) *The IPS System*. Actes COLING-92, pages 870-874