Couplage d'un étiqueteur morpho-syntaxique et d'un analyseur partiel représentés sous la forme d'automates £nis pondérés

Alexis Nasr, Alexandra Volanschi*
LATTICE-CNRS (UMR 8094)
Université Paris 7
{alexis.nasr, alexandra.volanschi}@linguist.jussieu.fr

Résumé - Abstract

Cet article présente une manière d'intégrer un étiqueteur morpho-syntaxique et un analyseur partiel. Cette integration permet de corriger des erreurs effectuées par l'étiqueteur seul. L'étiqueteur et l'analyseur ont été réalisés sous la forme d'automates pondérés. Des résultats sur un corpus du français ont montré une dimintion du taux d'erreur de l'ordre de 12%.

This paper presents a method of integrating a part-of-speech tagger and a chunker. This integration lead to the correction of a number of errors made by the tagger when used alone. Both tagger and chunker are implemented as weighted £nite state machines. Experiments on a French corpus showed a decrease of the word error rate of about 12%.

Mots-clefs – Keywords

Analyse morpho-syntaxique, analyse syntaxique partielle, automates £nis pondérés
Part-of-speech tagging, chunking, weighted £nite state machines

1 Introduction

L'étiquetage morpho-syntaxique constitue souvent une étape préliminaire à un certain nombre de traitements linguistiques plus poussés tels que l'analyse syntaxique totale ou partielle. Les processus d'étiquetage morpho-syntaxique reposent généralement sur l'hypothèse que la catégorie d'un mot dépend d'un contexte local, qui est réduit à la catégorie du mot ou des deux mots précédents, dans le cas d'étiqueteurs probabilistes fondés sur les modèles de Markov cachés (MMC). Cette hypothèse est généralement correcte et a permis la réalisation d'étiqueteurs ef£caces et précis (de l'ordre de 95% de mots correctement étiquetés) dont les paramètres sont estimés à partir d'un corpus annoté. Il demeure que cette hypothèse n'est pas toujours véri£ée

*Ce travail a été partiellement £nancé par le projet WATSON dans le cadre de l'action Technolangue.

et est à l'origine d'une partie des erreurs d'étiquetage. Ces dernières mènent généralement à des erreurs dans les traitements suivants, voire à leur échec, en particulier pour l'analyse syntaxique. Cette situation est particulièrement frustrante dans la mesure où les traitements syntaxiques possèdent souvent les connaissances qui auraient pu éviter les erreurs d'étiquetage. Le but de cet article est de pallier partiellement ce problème en couplant les deux étapes d'étiquetage et d'analyse partielle. Dans un tel couplage, le choix de la catégorie d'un mot est effectué en tenant compte des connaissances propres à l'étiqueteur, mais aussi de celles provenant de l'analyseur partiel.

Le type d'erreur que l'on vise à corriger peut être illustré par la phrase suivante : *La recapitalisation n'est pas indispensable.* Lors de l'étiquetage morpho-syntaxique de cette phrase, le choix de la catégorie correcte pour l'adjectif *indispensable* (adjectif quali£catif féminin singulier) est délicat du fait que ce dernier peut être féminin ou masculin et que le nom avec lequel il s'accorde (*recapitalisation*) est relativement éloigné de l'adjectif, du moins pour un étiqueteur probabiliste fondé sur un MMC. Dans un tel cas, un analyseur partiel regroupera respectivement les suites *la recapitalisation*, *n'est pas* et *indispensable* au sein d'unités appelées *chunks*. Le résultat de ce regroupement est le rapprochement des deux unités (*la recapitalisation* et *indispensable*) entre lesquelles s'effectue l'accord et la possibilité de le modéliser dans un MMC. Le modèle de couplage proposé ici se pose en alternative à un modèle séquentiel où l'analyseur partiel prend en entrée la meilleure solution de l'étiqueteur. Il n'est alors plus possible de revenir sur les choix effectués par ce dernier.

Cet article vise un autre objectif qui est de montrer l'avantage de réaliser ces traitements à l'aide d'automates £nis pondérés et d'opérations sur ces derniers. Dans ce cadre, toutes les données (phrase à analyser, lexique, grammaire, n-grams) sont représentées sous la forme d'automates et (quasiment) tous les traitements sont réalisés par des opérations standard de manipulation d'automates. Cette homogénéité possède plusieurs avantages dont le premier est la facilité de combiner différents modules entre eux grâce aux opérations de combinaison d'automates, combinaisons plus dif£ciles à réaliser lorsque les différents modules reposent sur des modèles formels différents. Un autre avantage de l'homogénéité de ce cadre est la facilité de mise en œuvre : plus de formats spéci£ques à concevoir pour différents types de données, plus d'algorithmes à adapter, à programmer et à optimiser. La réalisation de tels traitements dépend de manière cruciale de l'existence de bibliothèques logicielles de manipulation d'automates. Dans le cadre de ce travail, nous avons utilisé les outils FSM et GRM de ATT (8). Notre travail se situe dans la mouvance du traitement probabiliste de la langue à l'aide d'automates pondérés, dont on trouvera un apperçu dans (12). Il se distingue dans son esprit d'autres approches fondées sur les automates £nis non probabilistes, telles qu'INTEX (7), dans lesquelles des règles sont construites manuellement pour être ensuite utilisées dans le cadre de traitements automatiques.

L'organisation de l'article est la suivante : dans la partie 2, on reprend quelques dé£nitions concernant les automates pondérés et on introduit quelques notations. Les sections 3 et 4 décrivent respectivement les principes d'un étiqueteur probabiliste et d'un analyseur partiel et leur implémentation sous la forme d'automates pondérés. Dans la section 5, l'integration des deux modules est décrite. En£n, des experiences sont présentées dans la partie 6 et des travaux futurs sont annoncés dans la partie 7. La revue de la littérature n'a pas été regroupée dans une section, nous avons préféré établir des comparaisons avec d'autres travaux dans le cours de l'article.

2 Dé£nitions et notations

Dans la suite de cet article, nous manipulerons deux types d'automates £nis, des reconnaisseurs, qui permettent de reconnaître des mots u construits sur un alphabet Σ ($u \in \Sigma^*$) et des transducteurs, qui permettent de reconnaître des couples de mots (u, v) construits sur deux alphabets Σ_1 et Σ_2 ($(u, v) \in \Sigma_1^* \times \Sigma_2^*$). En plus des opérations régulières standard (union, concaténation et itération) dé£nies sur les deux types de machines, certaines opérations sont spéci£ques aux transducteurs, en particulier l'opération de *composition*, qui joue un rôle fondamental dans le reste de cet article. Etant donnés deux transducteurs A et B reconnaissant respectivement les couples de mots (u, v) et (v, w), la composition de A et B (notée $A \circ B$) est un transducteur qui reconnaît le couple (u, w).

On dé£nit de plus la notion de semi-anneau qui est un quintuplet $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ tel que $\mathbb{K}$ est un ensemble de scalaires muni de deux opérations généralement apellées addition (notée $\oplus$) et multiplication (notée $\otimes$) ayant chacune un élément neutre noté respectivement $\bar{0}$ et $\bar{1}$. En associant à chaque transition d'un reconnaisseur un poids prenant sa valeur dans un ensemble $\mathbb{K}$, on obtient un reconnaisseur pondéré construit sur un semi-anneau sur l'ensemble $\mathbb{K}$. Un reconnaisseur pondéré, en conjonction avec un semi-anneau $\mathbb{K}$ génère une fonction partielle qui associe aux mots du langage reconnu par le reconnaisseur des valeurs de $\mathbb{K}$. Etant donné un reconnaisseur R et un mot u, la valeur associée à u par R, notée $[\![R]\!](u)$, est le produit ($\otimes$) des poids des transitions du chemin de R correspondant à u. Si plusieurs chemins de R permettent de reconnaître u, alors $[\![R]\!](u)$ est égale à la somme ($\oplus$) des poids des différents chemins correspondant à u. Etant donné un reconnaisseur pondéré R, on dé£nit l'opérateur *n-meilleurs chemins*, noté $mc(R, n)$ qui retourne le reconnaisseur constitué de l'union des n chemins les plus probables dans R. Toutes ces notions sont étendues aux transducteurs.

Dans les expériences décrites dans ce papier on a associé aux transitions des transducteurs l'opposé de logarithmes de probabilités[1] ; on a utilisé le semi-anneau *tropical* sur $\mathbb{R}^+$. Dans ce dernier, l'opération $\otimes$ correspond à l'addition usuelle (pour connaître le poids d'un chemin on additionne les poids des transitions) alors que l'opération $\oplus$ est le minimum (le poids associé par un transducteur a un mot reconnu est le minimum des poids de tous les chemins du transducteur reconnaissant le mot, c'est-à-dire le chemin ayant la meilleure probabilité).

3 Etiquetage morpho-syntaxique

Le processus d'étiquetage morpho-syntaxique utilisé dans le cadre de ce travail reprend les principes de l'étiquetage morpho-syntaxique fondé sur les chaînes de Markov cachées, introduit dans (5). Les états du MMC correspondent aux catégories morpho-syntaxiques et les observables aux mots du lexique. Ces derniers constituent l'alphabet Σ_L et les étiquettes des catégories morpho-syntaxiques constituent l'alphabet Σ_C. Le processus d'étiquetage, dans un tel modèle, consiste à retrouver la suite d'états la plus probable étant donné une suite d'observables.

Les paramètres d'un MMC se divisent en probabilités d'émission et en probabilités de transition. Une probabilité d'émission est la probabilité d'un mot étant donné une catégorie ($P(m|c)$)

[1] On préfère les logarithmes de probabilités aux probabilités pour des questions de stabilité numérique (les probabilités pouvant être des réels très petits, on risque de ne pas pouvoir les représenter en machine). L'utilisation de l'opposé du logarithme permet d'obtenir les chemins de probabilité *maximale* lors de l'utilisation de l'opérateur *n-meilleurs chemins*.

tandis qu'une probabilité de transition est la probabilité qu'une catégorie x suive directement une catégorie y ($P(x|y)$). Ces deux ensembles de paramètres permettent de calculer la probabilité jointe d'une suite de catégories $c_{1,n}$ (une suite d'états du modèle) et d'une suite de mots $m_{1,n}$ (une suite d'observables) en utilisant les probabilités d'émission et de transition : $P(c_{1,n}, m_{1,n}) = P(c_1)P(m_1|c_1)\prod_{i=2}^{n} P(m_i|c_i)P(c_i|c_{i-1})$

Un tel modèle, appelé modèle *bigramme*, repose sur l'hypothèse markovienne qu'une catégorie ne dépend que de la catégorie précédente. Cette hypothèse, fort contraignante, peut être assouplie sans changer de cadre théorique en faisant dépendre une catégorie non plus de la catégorie précédente mais des deux catégories précédentes pour aboutir à un modèle *trigramme* qui est le modèle généralement utilisé pour une telle tâche. Dans un modèle trigramme, un état correspond non plus à une catégorie, mais à un couple de catégories.

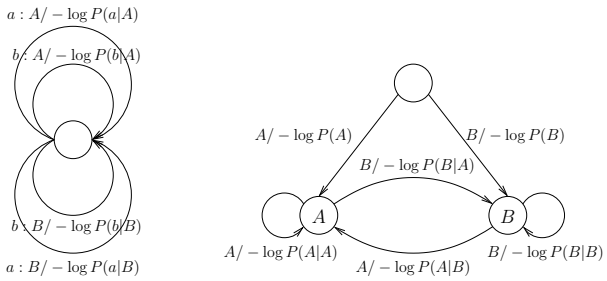

Figure1 : Les transducteurs E et T

Un tel MMC peut être représenté par deux transducteurs pondérés. Le premier, que nous appellerons E, et dont un exemple apparaît dans la partie gauche de la £gure 1 (dans cet exemple $\Sigma_L = \{a, b\}$ et $\Sigma_C = \{A, B\}$) permet de représenter les probabilités d'émission. Son alphabet d'entrée est Σ_L et son alphabet de sortie Σ_C. Ce transducteur est doté d'un seul état, et possède autant de transitions (de l'unique état vers lui même) qu'il y a de couples (m, c) où m est un mot du lexique ($m \in \Sigma_L$) et c une catégorie ($c \in \Sigma_C$) tels que la probabilité d'émission $P(m|c)$ soit non nulle. L'opposé du logarithme de cette probabilité ($-\log P(m|c)$) constitue le poids de la transition étiquetée (m, c). Dans la £gure 1, une telle transition est étiquetée $m : c/ - \log P(m|c)$. Le second transducteur, ayant pour alphabet d'entrée et de sortie Σ_C (partie droite de la £gure 1), appelé T, permet de représenter les probabilités de transition. Il reprend la structure du MMC : autant d'états que de catégories et des transitions entre tout couple d'états (x, y) (orienté de x vers y) tel que $P(y|x)$ est non nulle. Le poids de la transition est égal à $-\log P(y|x)$[2]. Dans le cas d'un modèle trigramme, la structure de l'automate T est plus complexe : un état correspond à une séquence de deux catégories et les poids des transitions sont de la forme $-\log P(z|xy)$.

La composition de E et de T ($E \circ T$) permet de combiner probabilités d'émission et de transition pour aboutir à un transducteur dont l'alphabet d'entrée est Σ_L et l'alphabet de sortie est Σ_C. Un tel transducteur permet d'associer au couple $(m_{1,n}, c_{1,n})$ le poids $[\![E \circ T]\!](c_{1,n}, m_{1,n}) = -\sum_{i=1}^{n} \log P(m_i|c_i) - \log P(c_1) - \sum_{i=2}^{n} \log P(c_i|c_{i-1})$ qui n'est autre que l'opposé du logarithme de la probabilité $P(c_{1,n}, m_{1,n})$, telle que dé£nie ci-dessus.

L'étiquetage d'une suite de mots particulière M est réalisé en représentant la suite M sous la forme d'un reconnaisseur de structure linéaire (une transition pour chaque mot de M), appelé lui-même M puis en effectuant la composition de M avec $E \circ T$. La recherche de la suite de catégories la plus probable étant donné M est alors réalisée par la recherche du meilleur chemin dans le transducteur $M \circ E \circ T$. L'étiqueteur s'écrit donc : $mc(M \circ E \circ T, 1)$

Les probabilités des trigrammes représentées dans l'automate T ne sont généralement pas

[2]Strictement parlant, l'automate décrit est un reconnaisseur, mais il peut être vu comme un transducteur dont l'alphabet de sortie est égal à l'alphabet d'entrée et dont chaque transition possède le même symbole en entrée et en sortie. Un tel transducteur représente par conséquent la relation identité réduite au langage reconnu par le reconnaisseur.

estimées par simple maximum de vraisemblance sur un corpus d'apprentissage, car des trigrammes apparaissant dans les textes à étiqueter peuvent n'avoir jamais été observés dans le corpus d'apprentissage. C'est la raison pour laquelle on a recours à des méthodes de lissage des probabilités, telles que les méthodes de repli (10) qui consistent à se *replier* sur la probabilité du bigramme `b c` lorsque le trigramme `a b c` n'a pas été observé dans le corpus et, lorsque le bigramme `b c` n'a pas été observé, à se replier sur l'unigramme `c`. Un modèle de repli peut être directement représenté sous la forme d'un automate comportant des transitions *par défaut* comme décrit dans (4). Etant donné un symbole α, une transition par défaut émanant d'un état q est empruntée lorsqu'il n'existe pas de transition émanant de q étiquetée par α. Dans le cas du modèle de repli, une transition par défaut est empruntée lorsqu'un trigramme ou un bigramme n'a jamais été observé. Il ne nous est pas possible ici de décrire plus en détail la structure de tels automates. Pour plus de détails, le lecteur est invité à se référer à l'article cité ci-desssus.

Plusieurs approches dans la littérature (14; 11; 9) utilisent les automates £nis pondérés a£n de simuler le fonctionnement d'un MMC. Dans les trois cas, les n-grammes sont représentés sous la forme d'automates, de manière proche de la notre. Cependant, ces travaux se distinguent du notre en ne modélisant pas directement les probablilités d'émission ($P(m|c)$) estimées sur un corpus d'apprentissage, mais en recourant à des classes d'ambiguïtés, qui sont des ensembles de catégories associées à un mot.

4 Analyse syntaxique partielle

L'analyse syntaxique partielle désigne un ensemble de techniques dont le but est de mettre au jour une partie de la structure syntaxique d'une phrase, plus précisément, la structure associée aux fragments qui n'ont qu'une analyse possible. Par exemple, même si une suite comme 'maison des sciences de l'homme' constitue dans une grammaire traditionnelle un groupe nominal ayant une structure complexe avec plusieurs niveaux intermédiaires, dans une analyse partielle elle sera segmentée en trois unités appelées *chunks* : [maison]$_{CN}$ [des sciences]$_{CP}$ [de l'homme]$_{CP}$ car le rattachement des syntagmes prépostionnels est potentiellement ambigu. Appelée aussi *chunking*, l'analyse partielle a été introduite par (3) comme réponse aux dif£cultés d'analyse soulevées par le traitement robuste des textes tout-venants.

Plusieurs approches dont (2) ont abordé l'analyse partielle à l'aide des automates £nis, plus précisément à l'aide des cascades de transducteurs £nis. Une cascade de transducteurs est une succession de transducteurs où chacun permet de reconnaître un type de chunk. L'entrée de chaque transducteur est constituée par la sortie du transducteur précédent. Notre solution consiste en l'application simultanée, plutôt que séquentielle, de tous les automates des chunks qui sont intégrés au sein d'un MMC.

Les chunks, du fait de leur caractère non récursif, peuvent être représentés sous la forme d'automates £nis construits sur l'alphabet Σ_C. A chaque type de chunk K (par exemple chunk nominal, prépositionnel, ...) correspond un automate appelé aussi K, qui reconnaît toute séquence de catégories qui constitue un chunk bien formé de type K. De plus, au chunk de type K sont associés deux symboles, un symbole de début de chunk, noté `<K>`, et un symbole de £n de chunk, noté `</K>`. L'ensemble des symboles de début et de £n de chunk constituent un nouvel alphabet appelé Σ_K. Les différents automates associés aux chunks sont regroupés entre eux au sein d'un transducteur, appelé A, qui constitue l'analyseur et dont la structure est représentée dans la £gure 2.

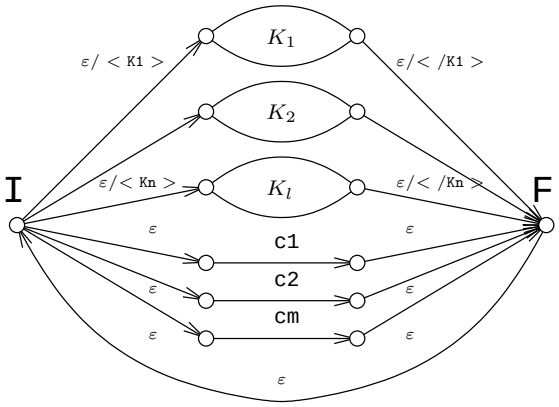

Figure 2 : Structure de l'analyseur partiel A

L'alphabet d'entrée de A est Σ_C et son alphabet de sortie est $\Sigma_C \cup \Sigma_K$. Il accepte en entrée des séquences de catégories et produit des séquences mélant catégories et symboles de début et de £n de chunk. Etant donné une séquence de catégories C en entrée, A produira en sortie la même séquence dans laquelle toute occurrence d'un chunk de type K sera encadrée des deux symboles <K> et </K>. A est composé de deux parties, une partie supérieure qui est elle même composée des différents automates de chunks, notés K_i mis en parallèle.

Les transitions reliant l'état initial de A aux états initaux des différents automates K_i permettent d'introduire les symboles de début de chunk et les transititions reliant les états d'acceptation des automates K_i à l'état F introduisent des symboles de £n de chunk. La partie inférieure de A est composée d'autant de transitions qu'il y a de catégories morpho-syntaxiques. En£n une transition ε reliant F à I permet de réaliser une boucle et de reconnaître ainsi plusieurs occurrences de chunks dans une séquence de catégories.

L'automate A reconnaît n'importe quel mot C construit sur Σ_C. L'analyse de C est réalisée en représentant C sous la forme d'un automate linéaire (une transition pour chaque catégorie constituant C) appelé lui aussi C et en effectuant la composition $C \circ A$. On pourra remarquer que le produit de cette composition est ambigu, car pour chaque sous-mot s de C correspondant à un chunk K_i, deux résultats seront produits : la reconnaissance de s en tant que chunk (passage à travers l'automate K_i) et la reconnaissance de s comme une suite de catégories ne constituant pas un chunk (passage dans les transitions de la partie inférieure de A). Parmi ces différents résultats, un seul nous intéresse, celui dans lequel toute occurrence de chunk a été marquée par l'introduction de balises de début et de £n de chunk. Il est facile de limiter le produit de la composition à ce seul résultat en associant à chaque transition intra chunk un poids de 0 et aux transitions extra chunk un poids de 1 et en ne gardant des résultats produits que le chemin de poids minimal. Le processus d'analyse peut être représenté par l'expression : $mc(C \circ A, 1)$

5 Couplage de l'étiquetage et de l'analyse partielle

Les modèles d'étiquetage morpho-syntaxique à l'aide des transducteurs pondérés cités dans la section 3, intégrent aussi (ou prévoient la possibilité d'intégrer) des contraintes syntaxiques dans le processus d'étiquetage. Kempe (11) prévoit la possibilité de composer la sortie du tagger avec des transducteurs encodant des règles de correction des erreurs les plus fréquentes, Tzoukerman (14) utilise des contraintes négatives a£n de diminuer de façon drastique la probabilité des chemins comportant des suites improbables d'étiquettes (par exemple un déterminant suivi d'un verbe). D'un point de vue général, notre travail se distingue des autres par le fait qu'il intègre deux modules complets (un module d'étiquetage et un module d'analyse partielle) au sein d'un seul, réalisant l'étiquetage et l'analyse partielle. Il ne s'agit pas d'intégrer dans un étiqueteur des grammaires locales conçues pour éliminer certaines structures agrammaticales, mais d'intégrer véritablement l'information statistique avec les connaissances linguistiques modélisées par l'analyseur partiel dans le but d'améliorer la qualité de l'étiquetage.

Le couplage de l'étiquetage morpho-syntaxique et du découpage en chunks peut être réalisé par simple composition des deux modèles que nous avons décrit : $mc(mc(M \circ E \circ T, 1) \circ A, 1)$. Ce modèle est une instance de l'architecture séquentielle que nous avons introduite et critiquée dans la section 1 : la sélection d'une étiquette morpho-syntaxique est réalisée indépendamment de la tâche d'analyse syntaxique (ici réalisée par un simple découpage en chunks) et ne peut être remise en cause par cette dernière.

Il est possible de fournir à l'analyseur non plus le meilleur étiquetage possible mais l'ensemble de toutes les solutions de l'étiqueteur représentées sous la forme d'un automate :$mc(M \circ E \circ T \circ A, 1)$. Ceci montre la souplesse du traitement par automates £nis. Mais un tel modèle n'offre pas beaucoup d'intérêt dans la mesure où l'analyseur n'a quasiment aucun pouvoir discriminant permettant de favoriser certaines des sorties de l'étiqueteur. En effet, contrairement à un analyseur fondé sur une grammaire hors-contexte, par exemple, qui n'associe une structure qu'aux phrases appartenant au langage reconnu par la grammaire, notre analyseur accepte toutes les suites de catégories, son rôle se borne à reconnaître certaines sous-suites de cette dernière comme formant des chunks. C'est la raison pour laquelle nous allons introduire une version probabiliste de l'analyseur partiel. Ce dernier effectue un découpage en chunks d'une suite de catégories et lui associe de plus une probabilité d'après un modèle dont les paramètres ont été estimés sur un corpus. Un tel modèle n'a pas pour objectif de favoriser un découpage en chunks d'une même suite de catégories plutôt qu'un autre (l'analyse en chunks est unique !). Son objectif est de fournir un moyen de comparer entre elles différentes séquences de catégories possibles pour une même phrase. Pour cela, l'analyseur partiel associe à toute séquence de catégories une probabilité qui est d'autant plus élevée que la séquence de catégories correspond à des séquences de chunks bien formés, agencés dans un ordre linéaire observé sur un corpus d'apprentissage. Cette approche partage plusieurs points communs avec les travaux de (6) qui utilisent eux aussi des transducteurs pondérés pour réaliser un analyseur partiel probabiliste. Cependant, dans leur cas, plusieurs découpages de la phrase en chunks sont possibles et l'objectif de l'analyseur est de fournir le découpage le plus probable. De plus, leur analyseur prend en entrée une séquence unique de catégories.

La probabilité d'une suite de catégories découpée en chunks est calculée à partir de deux types de probabilités : des probabilités intra chunk et des probabilités inter chunks. Une probabilité intra chunk est la probabilité qu'une suite de catégories $c_{1,k}$ constitue un chunk d'un type K_i. Cette probabilité est notée $P_I(c_{1,k}|K_i)$. Les probabilités inter chunk sont les probabilités conditionnelles d'occurrence d'un chunk d'un type donné, étant donnés les $n-1$ chunks ou catégories précédents (s'agissant d'une analyse partielle, certaines catégories de la suite analysée ne seront pas intégrées dans des chunks). La probabilité associée par l'analyseur à une suite de catégories est le produit des probabilités internes des chunks qui le composent et des probabilités externes de la séquence des chunks reconnus.

Etant donné la suite `<s> D N V D N P D A N </s>`[3]. Le découpage proposé par l'analyseur est : `C = <s> <CN> D N </CN> V <CN> D N </CN> <CP> P D A N </CP> </s>`

La probabilité associée à cette séquence est le produit de la suite des chunks reconnus (notée $P_E(\cdot)$), et des probabilités internes de chacun des chunks :

$P(\texttt{C}) = P_E(\texttt{<s> <CN> V <CN> <CP> </s>}) \times P_I(\texttt{D N|<CN>})^2 \times P_I(\texttt{P D A N| <CP>})$

Les probabilités internes sont estimées par maximum de vraisemblance sur un corpus d'apprentis-

[3]Où `D`, `N`, `V`, `P` et `A` sont les étiquettes correspondant respectivement aux catégories *déterminant, nom, verbe, préposition* et *adjectif*.

sage, comme nous le verrons en 5.1. La probabilité d'une suite d'étiquettes de chunks et d'étiquettes morpho-syntaxique est calculée à l'aide d'un modèle n-gram, appelé modèle externe, appris lui aussi sur un corpus, qui modélise la probabilité d'un chunk étant donné les $n-1$ chunks ou catégories précédentes. Dans le cas d'un modèle externe bigramme, la probabilité externe de C est calculée de la manière suivante :

$P_E(\texttt{C}) = P_E(\texttt{<CN>|<s>}) \times P_E(\texttt{V|<CN>}) \times P_E(\texttt{<CN>|V}) \times P_E(\texttt{<CP>|<CN>}) \times P_E(\texttt{</s>|<CP>})$

5.1 Construction du modèle et estimation de ses paramètres

L'estimation des paramètres du modèle externe et des modèles internes s'effectue en deux étapes à partir d'un corpus étiqueté. Lors d'une première étape, le corpus est analysé par l'analyseur partiel A. Le résultat de cette analyse est un nouveau corpus dans lequel des symboles de début et de £n de chunks ont été introduits. Deux objets sont produits à partir de ce corpus. D'une part toutes les suites de catégories correspondant à chaque type de chunk K_i et d'autre part un corpus *hybride* dans lequel toute occurrence de chunk a été remplacée par un seul symbole, matérialisant le chunk (ce symbole n'est autre que la marque de début de chunk). Le corpus hybride se présente donc sous la forme d'une séquence de catégories et de symboles de chunk, trace du chunk qui a été détecté à cet endroit. Le premier va servir à estimer les probabilités intra chunks et le second les probabilités inter chunks. Les différentes étapes de ce traitement sont représentées dans la £gure 3.

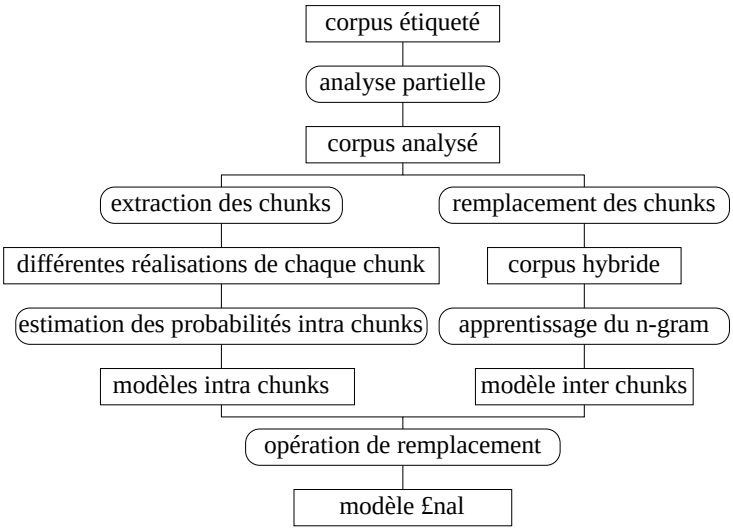

Figure 3 : Les étapes de la construction du modèle

L'estimation des probabilités inter chunk à partir du corpus hybride est identique à l'estimation des probabilités n-gram décrite en 3 et sur laquelle nous ne reviendrons pas. Ces probabilités sont représentées dans un transducteur (appelé modèle externe) reprenant la structure de T dans la £gure 1 et dont les transitions sont étiquetées par des catégories ou des symboles de chunks. L'estimation des probabilités intra chunk est une simple estimation par maximum de vraisemblance. Etant donné un chunk C_k et n suites différentes d'étiquettes (s_1, s_2, ... s_n) représentant toutes les réalisations de ce chunk dans un corpus d'apprentissage, on note n_i le nombre d'occurrences de la suite c_i. La probabilité de s_i n'est autre que sa fréquence relative : $P(s_i) = \frac{n_i}{\sum_{k=1}^{n} n_k}$. Cette probabilité est la probabilité du chemin correspondant à s_i dans le reconnaisseur K_i.

Les modèles intra chunks et le modèle externe sont combinés pour former un unique transducteur à l'aide de l'opération de remplacement, intoduite dans (13). Cette dernière permet de remplacer dans le modèle externe une transition <Ki> par l'automate K_i. Le transducteur résultant est appelé AP (pour analyseur probabiliste). Le modèle conjoint d'étiquetage et d'analyse est maintenant $mc(M \circ E \circ AP, 1)$.

6 Expériences

Les expériences ont été menées sur le corpus étiqueté Paris 7 (1). Le corpus est constitué de $900K$ mots étiquetés avec un jeu de 222 étiquettes indiquant la catégorie et les traits morphologiques des mots. On a réservé une partie du corpus de $760K$ mots pour l'apprentissage (*App*). Les tests ont été réalisés sur un fragment de $66K$ mots (*Test*). Le taux d'erreur du modèle trigramme (noté $\mathcal{M}_1$), tel qu'il est décrit dans la partie 3 sur *Test* est de $2,18\%$[4]. Ce chiffre constitue notre point de référence. 28 grammaires de chunks différents ont été construites manuellement. Ces grammaires appartiennent à une sous-classe des grammaires hors-contexte qui représentent des langages réguliers et qui peuvent être compilées sous forme d'automates a£n d'effectuer l'analyse partielle du corpus. Les probabilités intra chunks et les probabilités externes ont été estimées sur *App*. Les expériences ont été réalisées grâce aux librairies FSM et GRM de AT&T

Les performances du modèle $mc(M \circ E \circ AP, 1)$ (noté $\mathcal{M}_2$) sont quasiment identiques à celle de $\mathcal{M}_1$. Cependant, les deux modèles n'effectuent pas les mêmes erreurs. En effet, $\mathcal{M}_2$ corrige 30% des erreurs effectuées par $\mathcal{M}_1$ mais effectue quasiment autant d'erreurs en plus. Ces nouvelles erreurs ont différentes causes dont certaines proviennent de l'hypothèse du modèle $\mathcal{M}_2$ que la forme d'un chunk (la suite de catégories qui constituent le chunk) est indépendante du contexte d'occurrence de ce dernier. Cette hypothèse n'est pas toujours valide, comme l'illustre la phrase 'la discussion a été ouverte par l'article ...'. Dans cet exemple, 'ouverte' a correctement été étiqueté *participe passé* par $\mathcal{M}_1$, alors que $\mathcal{M}_2$ l'a étiqueté *adjectif*. La raison de cette erreur provient du fait que $\mathcal{M}_2$ a reconnu 'a été ouverte' comme chunk verbal et a choisi la catégorie de 'ouverte' indépendamment du contexte du chunk. $\mathcal{M}_1$ de son côté a tiré parti du fait que 'ouverte' était suivi d'une préposition pour lui assigner la catégorie *participe passé*. A£n de pallier partiellement ce problème, nous avons combiné les modèles $\mathcal{M}_1$ et $\mathcal{M}_2$ au sein du modèle suivant : $mc((M \circ E \circ AP) \cap (M \circ E \circ T), 1)$, noté $\mathcal{M}_3$. Ce dernier ne conserve que les solutions communes à $\mathcal{M}_1$ et $\mathcal{M}_2$ auxquelles il associe la somme des poids attibuées par $\mathcal{M}_1$ et $\mathcal{M}_2$ ($[\![\mathcal{M}_3]\!](x) = [\![\mathcal{M}_1]\!](x) + [\![\mathcal{M}_2]\!](x)$[5]). Cette combinaison permet d'attenuer l'hypothèse d'indépendance. En effet, la dépendance entre la forme d'un chunk et son contexte d'occurrence est partiellement modélisé par $\mathcal{M}_1$. Le taux d'erreur de $\mathcal{M}_3$ sur *Test* est de $1,92\%$ soit une diminution de $11,9\%$ par rapport à notre modèle de référence : $\mathcal{M}_1$. Une analyse d'erreurs a montré que $\mathcal{M}_3$ corrige $15,5\%$ des erreurs de $\mathcal{M}_1$ mais effectue $7,9\%$ de nouvelles erreurs. Les raisons des erreurs effectuées par $\mathcal{M}_3$ sont diverses, certaines proviennent toujours de l'hypothèse d'indépendance citée ci-dessus, d'autres sont dues à l'estimation des probabilités intra chunk (la probabilité qu'une séquence de catégories donnée constitue un chunk d'une nature donnée). Ces dernières sont en effet estimées par simple maximum de vraisemblance et attribuent par conséquent une probabilité nulle à une réalisation de chunk qui n'a jamais été observée dans *App*. Une forme de lissage de ces probabilités semble nécessaire. D'autres erreurs proviennent des limites théoriques du modèle et nécessiteraient pour être corrigées une analyse syntaxique complète.

[4]Ce résultat est supérieur au résultat de (14) (4% de taux d'erreur) sur le même corpus et avec le même jeu d'étiquettes. Cette différence provient, au moins en partie, du fait que nous avons travaillé sans mots inconnus : tous les mots de *Test* apparaissent dans le dictionnaire. Nous avons effectué cette hypothèse car l'objet de notre travail est d'étudier l'apport de l'analyseur partiel sur les performances d'un étiqueteur fondé sur les MMC et nous estimons que l'in¤uence des mots inconnus sera quasiment la même sur les différents modèles que nous avons testé.

[5]Contrairement aux modèles $\mathcal{M}_1$ et $\mathcal{M}_2$ les poids associés à une séquence de mots par $\mathcal{M}_3$ ne correspondent pas à des probabilités.

7 Conclusion

Le travail présenté dans cet article a montré que la prise en compte de connaissances syntaxiques, sous la forme d'une analyse partielle, permet d'améliorer le résultat d'un étiqueteur morpho-syntaxique. Il a aussi montré que les différentes étapes pouvaient être réalisées à l'aide d'automates pondérés. De nombreuses améliorations pourraient être apportées au modèle décrit, telles qu'une meilleure méthode d'estimation des probabilités intra chunk ainsi qu'une meilleure modélisation de l'in¤uence du contexte sur la réalisation d'un chunk.

Références

1. Anne Abeillé and Lionel Clément. A tagged reference corpus for french. In *Proceedings LINC-EACL*, Bergen, 1999.

2. S. Abney. Partial parsing via £nite-state cascades. In Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information, Prague, Czech Republic, pages 8–15., 1996.

3. Steven P. Abney. Parsing by chunks. In Robert C. Berwick, Steven P. Abney, and Carol Tenny, editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 257–278. Kluwer, Dordrecht, 1991.

4. Cyril Allauzen, Mehryar Mohri, and Brian Roark. Generalized algorithms for constructing statistical language models. In *41st Meeting of the Association for Computational Linguistics*, pages 40–47, Sapporo, Japon, 2003.

5. L. R. Bahl and R. L. Mercer. Part of speech assignment by a statistical decision algorithm. In *Proceedings IEEE International Symposium on Information Theory*, pages 88–89, 1976.

6. Kuang-Hua Chen and Hsin-Hsi Chen. Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation. In *Meeting of the Association for Computational Linguistics*, pages 234–241, 1994.

7. http://www.nyu.edu/pages/linguistics/intex/.

8. http://www.research.att.com/sw/tools/{fsm,grm}.

9. Bryan Jurish. A hybrid approach to part-of-speech tagging. Technical report, Berlin-Brandenburgishe Akademie der Wissenschaften, 2003.

10. Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987.

11. André Kempe. Finite state transducers approximating hidden markov models. In *ACL'97*, pages 460–467, Madrid, Spain, 1997.

12. Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2), 1997.

13. Mehryar Mohri. *Robustness in Language and Speech Technology*, chapter Weighted Grammars Tools: the GRM Library, pages 19–40. Jean-Claude Junqua and Gertjan Van Noord (eds) Kluwer Academic Publishers, 2000.

14. Evelyne Tzoukermann and Dragomir R. Radev. Use of weighted £nite state trasducers in part of speech tagging. *Natural Language Engineering*, 1997.