UN EDITEUR POUR LA CONSTRUCTION DE BANQUES D'ARBRES

Yves LEPAGE & ANDO Sin-Iti
ATR-ITL Laboratoires d'Interprétation et Télécommunication
Hikaridai 2-2, Seika-tyo, Soraku-gun, 619-02 Kyoto, Japon
{lepage,ando}@itl.atr.co.jp

Résumé

Nous présentons un outil qui permet d'accélérer la construction de banques d'arbres. C'est essentiellement un éditeur de textes et d'arbres avec des fonctionalités supplémentaires. Tout d'abord, l'édition d'arbres y est réellement simple et directe, a l'instar de l'édition de textes. En plus, il est possible de rechercher des phrases ou des arbres semblables, pour s'en inspirer dans la construction de nouvelles structures. Enfin, des structures correspondant à la nouvelle phrase entrée peuvent être construites par analogie avec les données déjà existantes de la banque d'arbres, et être proposées pour approbation ou modification.

INTRODUCTION

Une banque d'arbres (ou corpus arboré) est un corpus dont chaque phrase porte une description linguistique écrite à la main par un indexeur (en fait, un arbre). Certaines approches statistiques en analyse utilisent déjà des banques d'arbres, et l'intérêt de telles banques est indéniable en tant que ressources linguistiques. Mais la construction d'une banque d'arbres est une entreprise très lourde. Nous proposons donc un outil, un éditeur avec des fonctionalités en plus, afin d'accélérer cette tâche.

0.1 L'édition

Construire une banque d'arbres est délicat et prend du temps. De fait, la cohérence des données dans ces banques est très souvent contestable : des portions semblables de textes se voient attribuer des structures différentes. Il s'agit là d'un point particulièrement sensible si ces données sont utilisées pour entraîner des modèles statistiques. La relecture prend donc un temps énorme.

Nous proposons de voir la construction d'une banque d'arbres comme une suite d'opérations d'édition et de recherche :

- éditer une phrase nouvelle (l'entrer) ;
- rechercher des phrases semblables, avec leur structure associée, pour s'en inspirer dans la construction de la nouvelle structure linguistique ;
- éditer la nouvelle structure linguistique (la construire) ;
- rechercher des structures semblables afin de vérifier la cohérence avec les données précédentes.

Cette vue des choses a un bénéfique effet « boule de neige » : plus la banque est grosse, plus son extension est sûre et rapide.

0.2 Un éditeur

Nous avons implémenté un outil qui permet l'édition de phrases et de structures linguistiques. Dans une première partie, nous montrons rapidement en quoi l'édition d'arbres y est conviviale.

Mais cet outil permet aussi la recherche de phrases ou de structures semblables dans une banque existante. Les définitions classiques de distance d'édition entre phrases ou entre arbres, ainsi que l'algorithme, nouveau, de recherche approximative utilisé, sont décrits dans une deuxième partie.

En plus, cet outil peut présenter à l'indexeur des structures candidates pour les phrases nouvelles. Ces structures candidates sont obtenues par analyse par analogie, technique rendue possible grâce à une explication de ce que serait l'analogie en termes de distances d'édition. Cela fait l'objet de la troisième partie.

1 EDITION

1.1 Edition de textes

Sous notre outil, l'édition de textes s'effectue comme sous les éditeurs de textes modernes (SimpleText sous Macintosh, Word de Microsoft, Textedit sous OpenLook de Sun Microsystems, etc.). Les commandes couper / copier / coller sont bien sûr disponibles, ainsi que la sélection par double ou triple clic au moyen de la souris (pour les mots et les lignes respectivement). Evidemment, les inconditionnels du clavier ont des équivalents-clavier à leur disposition.

1.2 Edition d'arbres

Un certain nombre de visualiseurs d'arbres sont disponibles gratuitement, (comme celui de wxWindows de Ian Smith), mais leur défaut rédhibitoire est qu'ils ne permettent pas l'édition. Il existe certes quelques outils qui permettent l'édition, comme celui de l'Universiti Sains Malaysia, ou `xoobr` (de Mark Stern), mais elle y est particulièrement lourde (boîte de dialogue pour l'entrée d'une étiquette, addition de nouveaux nœuds uniquement par menu, etc.).

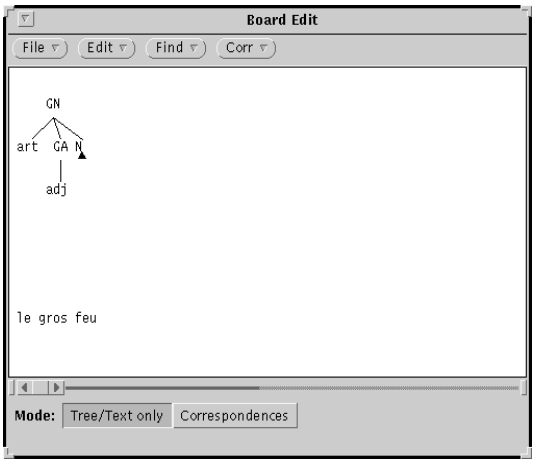

Figure 1: L'éditeur (version OpenLook)

Sous notre outil, l'édition d'arbre est simple et directe de la même façon que l'édition de textes est simple et directe, c'est-à-dire que l'interaction se fait directement dans le dessin de l'arbre, sans qu'il y ait menu ou boîte de dialogues.

Cela n'est possible que grâce à un parallèle rigoureux entre nœuds / sous-arbres d'un côté et mots / lignes de l'autre. Ce parallèle sous-tend toutes les fonctions d'édition d'arbres : toute fonction
(cliquer, sélectionner, insérer, couper / copier / coller, etc.) en édition d'arbres a exactement le même comportement que dans l'édition de textes. Ainsi, la séquence : `GN <retour> art <espace> GA<retour> adj <flêche haute><espace> N` produit l'arbre de la figure 1. Non seulement, la sélection directe par la souris est possible, mais encore l'insertion d'un frère ou d'un fils est possible par pointage directe avec la souris sur l'endroit où insérer, les branches apparaissant automatiquement.

1.3 Correspondances

Une fonctionalité particulière de notre outil est que des liens peuvent être établis entre le texte et l'arbre de façon à préciser quelle partie du texte correspond à quelle partie de l'arbre [Boitet and Zaharin 88]. Cela est très important pour les représentations non-projectives, comme certaines structures de dépendance.

La création des correspondances se fait par menu selon des schémas prédéfinis (projectives pour les structures en constituants, préfixe / infixe / postfixe pour les structures de dépendance). L'utilisateur peut ensuite raffiner grâce à d'autres commandes spécialisées (remise à zéro, décalage à gauche ou à droite). La visualisation des correspondances s'effectue en mode Correspondances par simple sélection. Toute sélection dans l'arbre (respectivement dans le texte) provoque la sélection simultanée de la partie correspondante dans le texte (respectivement dans l'arbre).

Dans ce mode, les commandes couper / copier / coller agissent simultanément sur le texte et sur l'arbre, grâce aux correspondances. Par exemple, couper le sous-arbre `GA` dans l'exemple précédent provoque aussi la suppression de `gros` dans le texte.

2 RECHERCHE

Lors de la construction d'une banque d'arbres, et après avoir édité une phrase nouvelle, l'indexeur a la possibilité de rechercher des phrases semblables, avec leur structure associée, afin de s'en inspirer pour créer la structure linguistique associée à la nouvelle phrase. «Semblable» signifie à une distance inférieure à une borne donnée, au sens des distances d'édition.

2.1 Distances d'edition

Les distances d'édition, d'abord proposées par [Levenshtein 65] et reprises ensuite dans maintes communications (cf. [Wagner & Fischer 74] ou [Selkow 77]), sont, sous certaines conditions, de vraies distances au sens mathématiques. Elles sont aussi intuitives, car elles répondent au problème de la correction : *quel est le nombre minimal d'opérations d'édition nécessaires pour transformer une chaîne donnée en une autre ?* Les opérations d'édition sont l'insertion, la suppression et la substitution. A partir de ces opérations, on définit une distance par un calcul de minimum lors de la lecture parallèle des deux chaînes. Pour ce faire, on affecte un poids à chacune des trois opérations d'édition, par exemple 1 pour simplifier.

2.1.1 Distance entre mots

Définition 1 (Wagner et Fischer) *Soit $\mathcal{V}$ un vocabulaire,* dist *une distance sur $\mathcal{V} \cup \{\varepsilon\}$,* dist *est étendue à $\mathcal{V}^*$ de la façon suivante :*

$$\forall\ (a,b)\ \in\ \mathcal{V}^2,\ \forall\ (u,v)\ \in\ (\mathcal{V}^*)^2,$$

$$\mathrm{dist}(a.u,b.v) \;=\; \min\left(\begin{array}{l}\mathrm{dist}(a,\varepsilon)+\mathrm{dist}(u,b.v),\\ \mathrm{dist}(a,b)+\mathrm{dist}(u,v),\\ \mathrm{dist}(\varepsilon,b)+\mathrm{dist}(a.u,v)\end{array}\right)$$

$$\begin{array}{rcll}\mathrm{dist}(a,b) & = & 0 & \text{si } a=b\\ & & 1 & \text{sinon (substitution)}\\ \mathrm{dist}(\varepsilon,b) & = & 1 & \text{(insertion)}\\ \mathrm{dist}(a,\varepsilon) & = & 1 & \text{(suppression)}\end{array}$$

Les opérations d'édition peuvent être définies sur les caractères d'un mot. Ainsi la distance entre *réaction* et *répressionnaire* est de 10 caractères.

2.1.2 Distance entre phrases

Une distance entre phrases peut aussi être définie en considérant que, de même qu'un mot est une chaîne de caractères, de même, une phrase est une chaîne de mots. Les opérations d'édition portent alors sur les mots, et la substitution d'un mot par un autre comptera pour 1, de même que la suppression ou l'insertion d'un mot quel qu'il soit. Avec cette définition, la distance entre les les deux phrases *le gros feu vire au vert* et *le gros voyant est vert* est de 3 mots.

2.1.3 Distance entre arbres

La définition précédente s'étend facilement aux forêts, et permet donc, par cas particulier, de définir une distance entre arbres.

Définition 2 (Selkow) *Soit $\mathcal{V}$ un vocabulaire,* dist *une distance sur $\mathcal{V}\cup\{\varepsilon\}$,* dist *est étendue à l'ensemble $\mathcal{F}$ des forêts étiquetées sur $\mathcal{V}$ de la façon suivante :*

$$\forall (a,b)\in\mathcal{V}^2,\ \forall(u,v,u',v')\in\mathcal{F}^4,$$

$$\mathrm{dist}(a(u').u,b(v').v) \;=\; \min\left(\begin{array}{l}\mathrm{dist}(a(u'),\varepsilon)+\mathrm{dist}(u,b(v').v),\\ \mathrm{dist}(a(u'),b(v'))+\mathrm{dist}(u,v),\\ \mathrm{dist}(\varepsilon,b(v'))+\mathrm{dist}(a(u').u,v)\end{array}\right)$$

$$\begin{array}{rcl}\mathrm{dist}(a(u'),\varepsilon) & = & \mathrm{dist}(a,\varepsilon)+\mathrm{dist}(u',\varepsilon)\\ \mathrm{dist}(a(u'),b(v')) & = & \mathrm{dist}(a,b)+\mathrm{dist}(u',v')\end{array}$$

$$\begin{array}{rcll}\mathrm{dist}(a,b) & = & 0 & \text{si } a=b\\ & & 1 & \text{sinon (substitution)}\\ \mathrm{dist}(\varepsilon,b) & = & 1 & \text{(insertion)}\\ \mathrm{dist}(a,\varepsilon) & = & 1 & \text{(suppression)}\end{array}$$

Cette définition signifie que les opérations d'édition s'appliquent aux nœuds des arbres. Ainsi, pour simplifier, sous notre éditeur, la suppression d'un nœud ou son insertion comptent pour 1, ainsi que la substitution d'une étiquette par une autre sur un nœud existant. Cette définition permet d'établir le résultat de la figure 2.

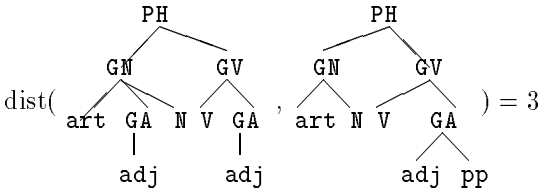

Figure 2: Distance entre arbres

2.2 Recherche approximative

Plusieurs algorithmes de *pattern-matching* approximatif, c'est-à-dire pour la recherche de patrons à une distance inférieure à un seuil donné d'une chaîne donnée, ont été proposés [Landau & Vishkin 88], [Wu & Manber 92].

2.2.1 Principe

Notre méthode se fonde sur une modification de l'algorithme de Aho & Corasick [Aho & Corasick 75]. Elle consiste a générer à l'avance l'ensemble des patrons situés à une distance inférieure au seuil du patron lui-même, en introduisant un caractère spécial pour coder n'importe quel caractère [Pinter 85]. Cet ensemble peut être réduit de façon appréciable en tenant compte des propriétés des chaînes recherchées. Il suffit en effet de générer les chaînes minimales au sens de l'inclusion.

Lorsque l'on fait varier le seuil, de 0 à la longueur du prototype, on observe alors une symétrie dans le cardinal de cet ensemble autour de la moitié de la longueur du prototype. Cette réduction considérable de l'explosion combinatoire n'apparaît pas dans les autres méthodes, où l'explosion croît avec le seuil.

L'ensemble des patrons une fois généré, une version non-déterministe de l'algorithme de Aho & Corasick tenant en compte le caractère spécial est facilement implémentable.

2.2.2 Algorithme

La méthode précédente de génération peut être reformulée en méthode de filtrage. En effet, la génération explicite des patrons n'est pas nécessaire. Il est équivalent de vérifier des contraintes sur des paramètres représentant les chaînes. On peut montrer que ces paramètres se réduisent à un triplet d'entiers pour chaque occurrence possible détectée.

Lors de l'interprétation de l'algorithme, à chaque caractère du texte d'entrée lu, l'ensemble des triplets de paramètres est remis à jour. Certains triplets ne vérifiant plus les contraintes disparaissent, et de nouveaux triplets, correspondant à de nouvelles occurrences possibles, apparaissent.

2.2.3 Expériences et mesures

L'algorithme précédent filtre exactement les phrases contenant des morceaux de phrases à une distance inférieure ou égale à un seuil donné. Il n'y a donc pas vraiment de sens à faire des tests de rappel / précision ou de silence / bruit. En revanche, il peut être intéressant de comparer cet algorithme avec l'un des algorithmes les plus rapides du moment, *agrep* [Wu & Manber 92].

Notre expérience a consisté à lancer la recherche approximative, pour chacun des 25 000 mots d'un dictionnaire, sur ce même dictionnaire, et ce pour tous les seuils d'erreurs possibles, de 1 à la longueur du mot moins 1. La longueur maximale d'un mot dans ce dictionnaire est de 22 lettres. On vérifie, bien entendu, que notre algorithme et *agrep* donnent les mêmes résultats.

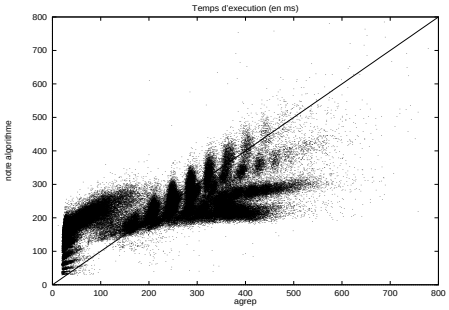

Figure 3: Comparaison entre *agrep* et notre algorithme

La figure 3 montre le nuage de points obtenu par la comparaison de notre algorithme avec *agrep*. Chaque point donne le temps d'exécution des deux algorithmes pour le même mot et le même seuil.

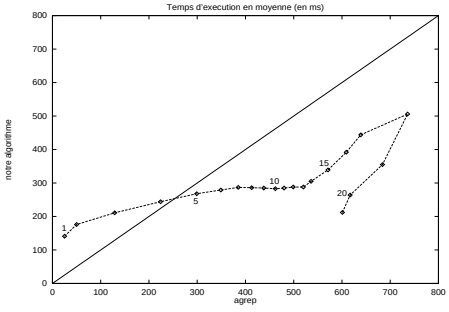

Figure 4: Comparaison en moyenne pour chaque seuil donné

La figure 4 compare les deux algorithmes en moyenne pour chaque seuil donné (de 1 à 21 en suivant la courbe de la gauche vers la doite). Notre algorithme est meilleur en moyenne, puisqu'on passe sous la diagonale assez rapidement. De plus, la propriété de notre algorithme mentionnée plus haut est mise en évidence : ses temps d'exécution décroissent plus vite que ceux de *agrep* lorsque le seuil d'erreurs croît.

2.3 Recherche de phrases semblables

L'algorithme précédent est utilisé pour la recherche de phrases semblables sous l'éditeur.

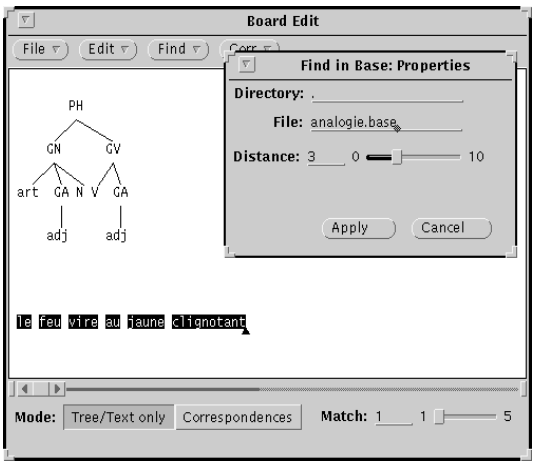

Figure 5: Résultat d'une recherche approximative

La figure 5 montre notre outil après recherche approximative (avec un seuil de 3 mots) pour la phrase *le gros feu vire au vert*. Une règlette en bas de l'éditeur permet de visualiser toutes les phrases trouvées, au nombre de 5 ici.

Il est à mentionner qu'une modification de l'algorithme précédent sur les chaînes permet la recherche approximative sur les arbres.

La fonction de recherche s'avère particulièrement utile pour une visualisation sélective d'une banque d'arbres. Ayant à notre disposition un extrait de la banque d'arbres de l'université de Pennsylvanie, notre tout premier exemple *from Dallas to Denver*, nous a immédiatement permis de mettre en évidence deux incohérences de codage dans cette banque. D'abord, le type de groupe prépositionnel précédent est codé de deux façons différentes : soit comme une séquence de deux sous-arbres de racine `PP`, soit comme un seul sous-arbre de racine `PP` dominant la représentation précédente. Ensuite, la ponctuation de fin de phrase apparaît dans la structure soit comme fils de la racine représentant la phrase, soit comme son frère.

3 ANALYSE PAR ANALOGIE

L'éditeur permet plus que la recherche de phases ou d'arbres semblables : il peut proposer, pour une phrase nouvelle, des analyses possibles obtenues par analogie.

3.1 L'analogie (Saussure)

3.1.1 Description

L'analogie [Saussure 16] est en quelque sorte la résolution d'une équation ou proportion entre mots : *étant données deux formes d'un même mot, et une forme seulement d'un second mot, la forme manquante peut être produite.*

fr.: *réaction* : *réactionnaire* = *répression* : x
x = *répressionnaire*

L'analogie est susceptible d'expliquer les paradigmes flexionnels, des conjugaisons aux déclinaisons, et sous-tend la productivité de la langue, c'est-à-dire le fait que des formes compréhensibles peuvent être produites, qui sont valides, mais ne sont pas répertoriées dans les lexiques usuels.

Elle explique aussi certaines fautes courantes dans la bouche des enfants.

fr.: *prendrai* : *prendre* = *viendrai* : x
x = *viendre*

3.1.2 Une possible explication

Plutôt que celle de proportions entre nombres, notre vue de l'analogie est géométrique. C'est celle d'un rectangle : les côtés opposés et les diagonales y ont, respectivement, mêmes longueurs (figure 6).

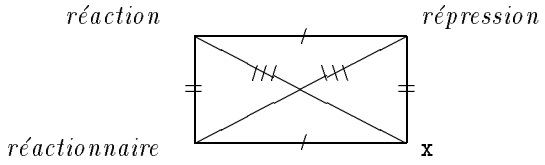

Figure 6: L'analogie comme rectangle

En terme de distances, on dira que la distance de n'importe quel terme à l'inconnue x est égale à la distance entre les deux termes restants. On pose donc l'équivalence suivante, dans laquelle les distances sont les distances d'édition mentionnées plus haut, portant sur des mots, des phrases ou des arbres indifféremment.

Définition 3 (Analogie)

$$u : v = w : \mathtt{x} \overset{\Delta}{\Longleftrightarrow} \begin{cases} \mathrm{dist}(u,v) = \mathrm{dist}(w,\mathtt{x}) \\ \mathrm{dist}(u,w) = \mathrm{dist}(v,\mathtt{x}) \\ \mathrm{dist}(v,w) = \mathrm{dist}(u,\mathtt{x}) \end{cases}$$

Cette définition rend compte partiellement de plusieurs phénomènes linguistiques. Tout d'abord, la transitivité est vérifiée sur nombre d'exemples linguistiques. Il s'agit du fait qu'à peu près n'importe quel représentant d'un groupe de conjugaison / déclinaison / etc. peut être choisi comme modèle du groupe. Ensuite, cette définition rend compte des phénomènes de préfixation / suffixation dans de nombreux cas linguistiques. C'est le cas de l'exemple ci-dessus avec *répressionnaire*. De même, l'infixation marche dans de nombreux cas, phénomène essentiel pour la description des langues sémitiques[1].

arabe: *arsala* : *mursilun* = *aslama* : x
x = *muslimun*

Nous ne prétendons pas donner ici la seule explication possible de l'analogie, car elle a de nombreux défauts. En particulier, les propriétés mentionnées ne sont pas vérifiées mathématiquement en général. Mais certaines conditions nécessaires pour qu'elles deviennent vraies, et que nous avons mises en évidence par ailleurs, semblent acceptables linguistiquement.

3.1.3 Expériences et mesures

Fermeture par analogie A partir d'un corpus donné, il est possible de générer automatiquement par analogie un ensemble de nouvelles phrases. En effet, pour tout triplet de phrases du corpus considéré, la résolution de l'analogie peut éventuellement fournir une phrase nouvelle.

Cette expérience a été réalisée sur un tout petit corpus exemple de 31 phrases, traduction des phrases japonaises et anglaises du corpus artificiel de Sato [Sato 91] pour ses expériences d'apprentissage automatique de grammaires non-contextuelles.

je suis Takuma.
je suis Taro.
tu es Takuma.
tu es Hanako.
je suis un garçon.
tu es une fille.
tu es un garçon.
je suis un grand garçon.
...

Le résultat est surprenant et extrêmement prometteur, puisque 92 phrases nouvelles sont générées[2], parmi lesquelles 41 s'avèrent grammaticalement correctes !

tu es Taro.
cela est le chien de Takuma.
je suis Hanako.
**jtulle est le profesuiur de Hanako.*
**jtulle est le profesuiur de Takuma.*
elle n'est pas la mère de Takuma.
il est Hanako.

[1] *arsala* (il envoya) et *aslama* (il se convertit) verbes à la 3ème personne du singulier masculin au passé; *mursilun* (expéditeur) et *muslimun* (un converti) noms d'agent.

[2] Dans cette expérience, l'analogie est résolue en considérant les phrases comme des chaînes de caractères, et non comme des chaînes de mots.

**tmo es Hanako.*
...

On peut alors ajouter ces phrases nouvelles au corpus pour obtenir la *fermeture par analogie*. Et l'on peut envisager de recommencer indéfiniment pour calculer la fermeture transitive du corpus par analogie.

Prégnance Nous proposons d'appeler *prégnance théorique* d'un corpus le rapport entre le nombre total de phrases générées par analogie et le nombre de phrases du corpus. On appellera de même *prégnance réelle* le rapport entre le nombre de phrases générées par analogie grammaticalement correctes et le nombre de phrases du corpus.

Notre petit corpus artificiel a une prégnance théorique de 297%, et une prégnance réelle de 135%. Une expérience sur un corpus réel verrait certainement ces chiffres diminuer considérablement.

Densité Dans une seconde expérience, nous avons calculé toutes les distances entre phrases (respectivement entre arbres) d'une même banque d'arbres.

La banque d'arbres utilisée est un extrait de la banque d'arbres de l'université de Pennsylvanie, comprenant 787 phrases, chacune avec son arbre correspondant. Théoriquement, il peut exister 787 $\times$ 786 $\times$ 785 $\times$ 784 quadruplets dont un certain nombre seulement vérifie la relation d'analogie[3].

Nous proposons d'appeler *densité analogique* d'un corpus le rapport entre le nombre de quadruplets en relation d'analogie et le nombre théorique de quadruplets.

Le corpus de notre expérience a une densité de 0,1% pour les phrases comme pour les arbres.

3.2 Analyse par analogie

L'approche *par l'exemple* en traduction automatique [Nagao 84] repose sur l'idée que, si deux phrases sont «proches», alors leurs analyses seront aussi, sans doute, «proches». Par conséquent, si on connaît l'analyse de l'une des phrases, on devrait pouvoir obtenir l'analyse de la seconde en y appliquant de légères «modifications» [Sadler and Vendelmans 90]. Il s'agit pour nous de donner consistance à ce vocabulaire éminemment flou : «proche» signifiera à une certaine distance au sens des distances d'édition, et «modification» signifiera opération d'édition.

3.2.1 Principe

L'analogie est un moyen de calculer automatiquement ces «proximités» et les «modifications» nécessaires. En effet, supposons qu'une phrase nouvelle, φ, soit ajoutée à la banque d'arbres, et supposons que deux phrases puissent être retrouvées par recherche approximative, alors nous sommes en possession des trois membres d'une analogie qui peut être résolue pour obtenir la valeur de la phrase inconnue `x`. Il est parfois possible d'obtenir une configuration où les trois derniers membres de l'analogie, y compris la solution `x`, se trouvent parmi le nombre important de phrases retrouvées dans la banque par recherche approximative (voir figure 7).

Dans ce cas, puisque les trois phrases sont éléments de la banque d'arbres, leurs analyses s'y trouvent aussi. Nous avons donc trois arbres à notre disposition, qui constituent à leur tour les termes d'une analogie à résoudre (figure 8). On posera que l'arbre α, solution de l'analogie, est l'analyse de la phrase φ, ce qui reste évidemment à être vérifié par l'indexeur. Si cela ne s'avère pas, l'indexeur a les outils d'édition à sa disposition pour effectuer les corrections nécessaires.

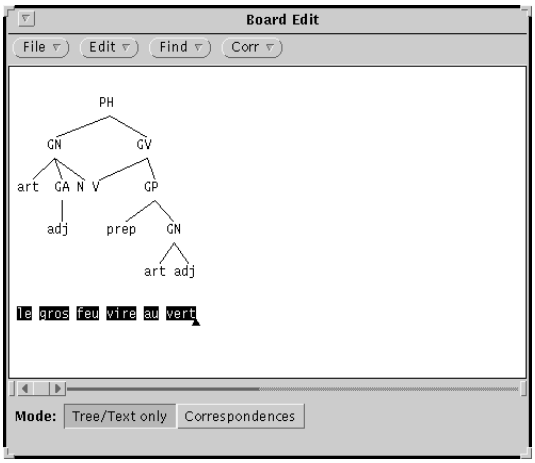

Figure 9: Analyse par analogie

La figure 9 montre le résultat du calcul par analogie pour la phrase *le gros feu vire au vert*, obtenu en cliquant sur le choix Analogie du menu Corr de notre éditeur.

Un mot sur l'utilité des correspondances. D'ordinaire, les systèmes de traitement de la langue fonctionnent au niveau des phrases. Ici, la recherche est possible pour une fraction seulement de phrase. Dans l'affichage des réponses, seule la partie pertinente des phrases est sélectionnée grâce aux correspondances. En conséquence, l'analyse par analogie est possible aussi sur des fractions de phrase.

3.2.2 Génération

La génération de phrases à partir d'un arbre serait aussi théoriquement possible, selon le même schéma, car notre modèle est strictement bidirectionnel et parce que les opérations (distances)

[3] Dans cette expérience, les distances sont la distance entre phrases considérées comme chaînes de mots et la distance entre arbres définie plus haut.

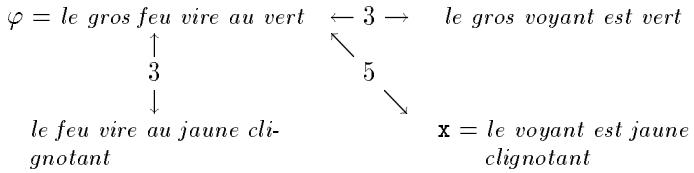

Figure 7: Nouvelle phrase, φ, et phrases obtenues par recherche approximative, en relation d'analogie. Distances mentionnées en mots.

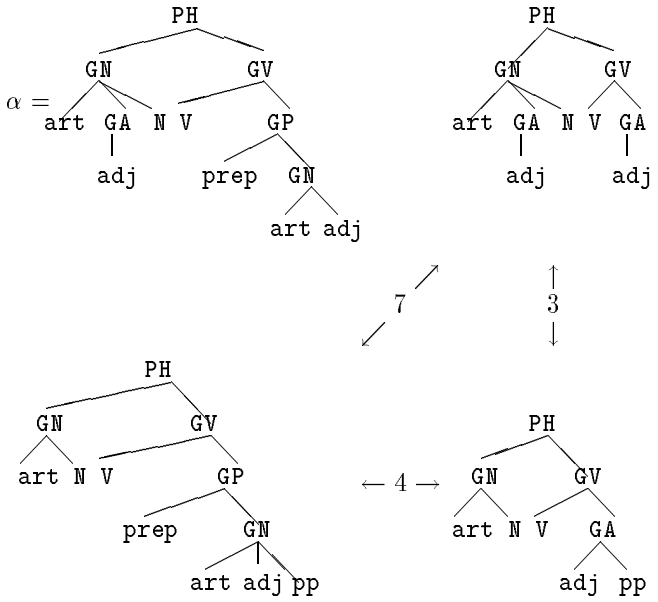

Figure 8: Analyses trouvées dans la banque d'arbres (les trois arbres en bas et à droite) et α, solution de l'analogie, analyse de la phrase φ. Distances mentionnées en nœuds.

sur les structures de données (textes et arbres) sont en fait implémentées comme une unique opération sur une structure de données généralisée (forêts).

3.2.3 Expériences et mesures

Afin d'avoir une idée plus précise de la puissance du procédé d'analyse par analogie, quelques mesures ont été effectuées sur la même banque d'arbres de 787 éléments. Pour tous les quadruplets possibles de phrases en relation d'analogie, (avec la restriction que les distances soient inférieures à la moitié des longueurs des mots considérés), nous avons calculé l'analyse de la première phrase par analogie. Nous rappelons que cette technique peut fournir zéro, une ou plusieurs solutions pour une même phrase.

Non-déterminisme Connaissant le nombre total de quadruplets donnant une analyse par analogie pour la première phrase du quadruplet, on considère le nombre de couples (première phrase, analyse) différents. Le rapport de ces deux nombres donne une mesure du *non-déterminisme* de la méthode, c'est-à-dire du nombre moyen d'analogies par lesquelles une même analyse peut être calculée.

Dans notre expérience, ce chiffre, extrêmement élevé, est de 103,8. Il signifie que l'obtention d'une analyse doit peu au hasard, puisque, pour une analyse donnée, il existe, en moyenne, avec cette banque d'arbres, plus de 100 façons différentes de la calculer.

Rappel En recherche documentaire, le *rappel* est défini comme le nombre de documents pertinents retrouvés, divisé par le nombre total de documents pertinents existant dans la base. Ici, ce sera donc le nombre de fois où la structure calculée par analyse par analogie est égale à la structure contenue dans la banque, divisé par le nombre de couples de phrases possédant la même structure.

Dans notre expérience, le rappel s'élève à 0,69.

Ce chiffre assez important montre que la technique est bien prometteuse.

Précision En recherche documentaire toujours, la *précision* est définie comme la proportion de documents répondant exactement à la requête considérée parmi tous les documents retrouvés. Ici, nous calculerons le nombre de fois où la structure calculée par analyse par analogie est égale à la structure contenue dans la base, divisé par le nombre de solutions proposées.

Dans notre expérience, la précision s'élève à 0,43. Cela signifie que, dans presque la moitié des cas, une analyse calculée par analogie est exacte.

Des mesures complémentaires de distance (au sens de la distance d'édition) permettent d'avoir une idée plus précise des différences entre les structures obtenues par analyse par analogie et les structures exactes.

Le maximum des différences est de 10 nœuds pour une taille des arbres de $9,8 \pm 5,41$ nœuds. La moyenne de ces différences est de 1,61 nœud seulement, avec une écart-type de 1,86 sur le total des réponses obtenues. Ces résultats sont assez bons, puisqu'il signifient qu'en moyenne, il suffit d'éditer moins de deux nœuds pour obtenir la structure exacte !

CONCLUSION

La construction d'une banque d'arbres destinée à des applications de traitement automatique des langues peut tirer bénéfice d'un effet «boule de neige», si la recherche de phrases ou d'arbres semblables est rendue possible. Cela permet d'accélérer la construction de la banque en même temps que d'améliorer la cohérence des données. Nous avons présenté un veritable éditeur convivial, tant pour l'édition de textes (ici, rien d'original) que pour l'édition d'arbres (chose nouvelle), qui permet non seulement la recherche approximative citée plus haut, mais propose aussi, lorsque cela est possible, des analyses éventuelles, obtenues par analogie, pour les nouvelles phrases entrées dans la banque.

BIBLIOGRAPHIE

[Aho & Corasick 75] Alfred V. Aho and Margaret J. Corasick
Efficient String Matching: An Aid to Bibliographic Search
Communications of the ACM, Vol. 18, No. 6, June 1975, pp. 333-340.

[Boitet and Zaharin 88] Christian Boitet and Zaharin Yusoff
Representation trees and string-tree correspondences
Proceedings of COLING-88, Budapest, 1988, pp 59-64.

[Landau & Vishkin 88] Gad M. Landau and Uzi Vishkin
Fast String Matching with k Differences
Journal of Computer and System Sciences, Vol. 37, 1988, pp. 63-78.

[Levenshtein 65] V.I. Levenshtein
Binary codes capable of correcting deletions, insertions and reversals
Dokl. Akad. Nauk SSSR, vol. 163, No. 4, August 1965, pp. 845-848.
English translation in *Soviet Physics-doklady* vol. 10, No. 8, February 1966, pp. 707-710.

[Nagao 84] Nagao Makoto
A Framework of a Mechanical Translation between Japanese and English by Analogy Principle
in *Artificial & Human Intelligence*, Alick Elithorn and Ranan Banerji eds., Elsevier Science Publishers, NATO 1984.

[Pinter 85] Ron Y. Pinter
Efficient string matching with don't care patterns
in A. Apostolico and Z. Galil (eds) *NATO Series, vol. F12, Combinatorial Algorithms on Words*, Springer Verlag, Berlin Heidelberg, 1985, pp. 11-29.

[Sadler and Vendelmans 90] Victor Sadler and Ronald Vendelmans
Pilot implementation of a bilingual knowledge bank
Proceedings of COLING-90, Helsinki, 1990, vol 3, pp. 449-451.

[Saussure 16] Ferdinand de Saussure
Cours de linguistique générale
publié par Charles Bally et Albert Sechehaye, Payot, Lausanne et Paris, 1916.

[Sato 91] Sato Satoshi
Example-based Machine Translation
Ph. D. thesis, Kyoto University, September 1991.

[Selkow 77] Stanley M. Selkow
The Tree-to-Tree Editing Problem
Information Processing Letters, Vol. 6, No. 6, December 1977, pp. 184-186.

[Wagner & Fischer 74] Robert A. Wagner and Michael J. Fischer
The String-to-String Correction Problem
Journal for the Association of Computing Machinery, Vol. 21, No. 1, January 1974, pp. 168-173.

[Wu & Manber 92] Sun Wu & Udi Manber
Fast Text Searching Allowing Errors
Communications of the ACM, Vol. 35, No. 10, October 1992, pp. 83-91.