

VERS UN FEEDBACK PLUS INTELLIGENT, LES ENSEIGNEMENTS DU PROJET FREETEXT

Sébastien L'HAIRE

Laboratoire d'Analyse et de Technologie du Langage

Université de Genève

4, pl. de l'Université

CH-1211 Genève 4

lhaire@lettres.unige.ch

Résumé : Cet article présente les enseignements apportés par le projet européen FreeText, qui a développé un prototype de logiciel d'apprentissage du français langue étrangère. Nous présentons les techniques de détection d'erreurs implémentées lors du projet. Puis nous décrivons les diverses sorties que nous proposons aux apprenants. Dans la dernière partie, nous envisageons certaines améliorations de l'interface, nous suggérons une série de nouveaux outils et nous montrons comment la sortie fournie par notre analyseur pourrait être utilisée dans un logiciel intelligent d'apprentissage des langues.

Mots-clés : ALAO, TALN, diagnostic, feedback, rétroaction, interface, XML

1. Introduction

Les besoins en logiciels d'Apprentissage des Langues Assisté par Ordinateur (ALAO) sont de plus en plus grands, que ce soit comme aide à l'enseignement en classe ou comme outil d'apprentissage autonome. Les applications de Traitement Automatique du Langage Naturel (TALN) peuvent constituer un apport important à l'apprentissage. Les produits commerciaux actuels disposent souvent d'aides à la prononciation basées sur des techniques basiques de reconnaissance vocale. Toutefois, ces techniques s'avèrent limitées et leur utilité pédagogique est contestée.

En outre, dans le domaine de l'écrit, il n'existe pas, à notre connaissance, d'application commercialisée comportant des outils de diagnostic basés sur le TALN. Les exercices proposés par la plupart des logiciels utilisent des techniques de reconnaissance de forme (*pattern matching*), qui limitent les variétés d'interaction et ne fournissent pas un feedback d'une qualité suffisante. Pour progresser à l'écrit, un apprenant doit produire des phrases complètes, qui ne peuvent être corrigées, à notre sens, que grâce à une analyse syntaxique. Cependant, les outils de TALN sont très sensibles aux erreurs et le traitement de celles-ci n'est pas une tâche aisée. Pour l'apprentissage de la grammaire, il est nécessaire d'avoir une rétroaction détaillée et fiable, qui n'induit pas en erreur les apprenants. C'est pourquoi Tschichold (2003) estime qu'actuellement, les outils produits par les équipes de recherche ne sont pas encore suffisamment fiables et qu'il faut se limiter au traitement d'erreurs locales. Pourtant, dans le même numéro spécial de la revue Calico où paraissent ses observations, des applications récentes présentent des résultats prometteurs.

Dans cet article, nous présentons nos travaux dans le projet d'enseignement du Français Langue Etrangère (FLE) **FreeText**¹ et présentons des perspectives de recherche. Nous commençons par présenter le projet **FreeText** à la section 2. A la section 3, nous décrivons les techniques de diagnostic mises en œuvre dans ce cadre. Les différentes interfaces développées au cours du projet sont montrées à la section 4. Nous poursuivons par une description de nos recherches actuelles et proposons diverses mises à profit de nos outils à la section 5, notamment en utilisant les techniques d'apprentissage *intelligent* assisté par ordinateur. Enfin une brève conclusion termine cet article à la section 6.

2. Le projet FreeText

Les recherches présentées dans cet article ont été menées dans le cadre du projet **FreeText**, qui avait pour but de développer un logiciel d'ALAO pour apprenants du FLE intermédiaires à avancés. **FreeText** contient quatre tutoriels présentant seize documents authentiques, allant du texte aux fichiers audiovisuels, qui illustrent différents actes communicatifs. Les exercices qui exploitent ces documents sont basés sur l'étude du corpus d'apprenants FRIDA

¹ Le projet **FreeText** relève du programme IST du 5^e programme-cadre de la Commission Européenne, contrat IST-1999-13093. Le volet suisse du projet **FreeText** reçoit le soutien financier de l'Office Fédéral de l'Education et de la Science, contrat 99.0049. Le contenu de cet article n'engage que ses auteurs et ne représente pas l'opinion de la Communauté Européenne. La Communauté Européenne n'est pas responsable de l'usage qui pourrait être fait des données figurant dans cet article. Les informations de ce document sont présentées telles quelles et aucune garantie n'est donnée que ces informations sont adéquates pour n'importe quelle utilisation particulière. Dès lors, l'utilisateur emploie ces informations à ses propres risques.

(Granger, 2003), dans le but de traiter des erreurs réellement commises par le public cible. Le logiciel dispose d'une grammaire de référence écrite spécialement pour le logiciel. De plus, **FreeText** utilise des outils de TALN, comme un visualiseur de phrases (v. section 4), un synthétiseur vocal et un système de diagnostic d'erreurs, afin d'offrir une rétroaction intelligente pour des exercices de production (relativement) libre.

Le consortium **FreeText** regroupe des partenaires dans différents domaines d'expertise de quatre pays européens: (i) le Department of Language Engineering, University of Manchester Institute of Science and Technology (UMIST), Royaume Uni, chargé du contenu didactique du logiciel ; (ii) le département de linguistique de l'Université de Genève, Suisse, qui développe des outils de TALN ; (iii) le Centre for English Corpus Linguistics (CECL), Université Catholique de Louvain, Belgique, responsable du travail autour du corpus et des différentes phases de validation ; (iv) Softissimo SA, Paris, France, pour l'intégration des différentes composantes dans une interface ergonomique.

FreeText est un système ouvert, où les enseignants peuvent introduire de nouveaux exercices et documents afin de répondre aux besoins particuliers de leurs apprenants. Le projet a démarré en avril 2000 et a pris fin en mars 2003. De plus amples informations sont disponibles sur le site du projet **FreeText** (2002).

3. Techniques de diagnostic

Dans cette section, nous présentons les techniques de diagnostic d'erreur qui ont été implémentées pour le logiciel **FreeText**. Les apprenants peuvent utiliser librement les outils, tant dans le cadre d'exercices que pour une utilisation indépendante. Le diagnostic se passe en deux étapes, d'abord une correction orthographique² puis un diagnostic d'erreurs syntaxiques. Le traitement est interrompu en cas d'erreur et l'apprenant peut corriger la phrase et la soumettre à nouveau aux outils.

Le système de diagnostic est basé sur l'analyseur FIPS (Wehrli, 1997), qui est basé sur la théorie des Principes et Paramètres (Chomsky & Lasnik, 1995) et qui fournit une analyse très riche et motivée linguistiquement. Par exemple, les mouvements de structure profonde à structure de surface sont marqués par l'insertion de traces (mouvement du passif, liens pronom relatif-antécédent etc.). FIPS est un analyseur modulaire à large couverture. En cas d'échec de l'analyse, il présente des résultats partiels sous forme de constituants bien formés, ce qui indique soit une structure agrammaticale, soit une construction qui dépasse la couverture de l'analyseur. FIPS a été adapté en implémentant deux techniques de diagnostic que nous survolons à présent et qui sont détaillées dans (Vandeventer Faltin, 2003 ; L'haire & Vandeventer Faltin, 2003a, 2003b).

La première technique que nous utilisons est celle bien connue du *relâchement de contraintes*. Dans un analyseur syntaxique, la combinaison de syntagmes n'est possible que si certaines règles sont satisfaites, comme les règles d'accord. Si aucune analyse complète n'est obtenue, on peut relâcher une ou plusieurs contraintes et marquer le type et le lieu de l'erreur dans la

² La correction orthographique est effectuée par une version adaptée du correcteur orthographique Hugo 2000 (2000). Des erreurs d'apprenants typiques du type *allerons* → *irons* ont été prises en compte.

structure, afin de fournir un diagnostic par la suite. Prenons une phrase authentique tirée du corpus FRIDA³ (Granger, 2003) :

(1) * L'héritage du passé est très forte et le sexisme est toujours présent.

Le syntagme adjectival « très forte » doit s'accorder avec le sujet de la phrase, « l'héritage du passé ». La vérification de l'accord se fait en prenant l'intersection des ensembles de traits des deux syntagmes. Ici, on obtient un ensemble vide pour le trait de genre. En activant le relâchement de contraintes, on procède à l'union des ensembles de traits de genre et l'on marque l'erreur dans la structure.

La seconde technique est appelée *réinterprétation phonologique*. Il est courant que les apprenants confondent des homophones, c'est-à-dire deux mots ayant une prononciation identique ou similaire. Notre analyseur fournit plusieurs morceaux d'analyse en cas d'absence d'analyse complète. Cette technique consiste alors à remplacer un mot dans la phrase par les homophones possibles afin d'obtenir une analyse complète. A cette fin, on remplace les mots aux frontières des morceaux d'analyse par un ou plusieurs homophones cherchés grâce à un lexique phonétique. Seuls les mots de catégorie différente ou ayant des traits différents sont pris en compte. Le processus d'analyse est alors relancé en prenant compte de toutes les combinaisons possibles. Examinons en détail le traitement de la phrase authentique mais simplifiée en (2), tirée du corpus FRIDA.

(2)a. *[Il faut maître] [les enfants au monde]. b. il faut mettre les enfants au monde.

Les crochets en (2a) distinguent les deux morceaux d'analyse obtenus lors d'une première analyse. Les mots à réinterpréter sont *maître* et *les*. *Les* n'a pas d'homophones mais est toujours dans le graphe de l'analyseur comme déterminant et comme pronom clitique. En revanche, pour *maître*, on obtient *maîtres*, *mettre*, *mètre* et *mètres*. La lecture de *mètre* comme substantif est exclue car elle a la même catégorie et les mêmes traits que *maître*. *Mètres* et *mettre* comme verbes (ind/sbj. prés, 2^e/3^e personne) et *maîtres* et *mètres* comme substantifs ne donnent aucune analyse. Par contre, *mettre* comme verbe infinitif donne une analyse complète.

Voici les erreurs traitées par notre système de diagnostic. Les erreurs sont affichées en gras ou par un soulignement. Toutes les erreurs sont traitées par relâchement de contraintes, sauf la catégorie des homophones.

Catégorie d'erreur	Acronyme	Exemple
Classe	CLA	Ce montre le problème.
Auxiliaire	AUX	Il a tombé.
Genre	GEN	Elle est beau .
Nombre	NBR	Les souris ont été mangée <u>_</u> .
Personne	PER	Je peut partir.

³ Le corpus FRIDA est un corpus d'apprenant du FLE récolté au CECL dans le cadre du projet **FreeText**. Les erreurs d'apprenants ont été codées d'après le domaine de l'erreur (syntaxe, forme, etc.), le type d'erreur (genre, diacritique, etc.) et la catégorie lexicale. 300 000 mots ont été ainsi codés et fournissent des informations sur les contextes spécifiques des erreurs. Les informations recueillies ont permis de cibler les exercices sur les domaines problématiques et de déterminer les erreurs sur lesquelles cibler le développement du système de diagnostic.

Catégorie d'erreur	Acronyme	Exemple
Voix	VOI	Il _ révolte contre la société.
Euphonie – élision	EUFE	ce aspect
Euphonie – contraction	EUFC	de le chat
Euphonie – t analogique	EUFT	A- _ il mangé?
Complémentation de l'adjectif	CPA	Il est capable à faire cela.
Complémentation du verbe	CPV	Il est tombé _ la chaise.
Ordre	ORD	Ils ont les vus.
Ordre de d'adverbe	ORDAV	C'est lui plutôt qui chasse
Ordre de l'adjectif	ORDAJ	un arabe pays.
Redondance	RED	Il parle beaucoup des langues.
Oubli de ponctuation	OUB	Peut _ il partir?
Négation	NEG	C' _ est pas un problème.
Homophonie	HOM	Elles peuvent maître au monde des enfants.

Tableau 1 - catégories d'erreur traitées par le système de diagnostic

Le principal inconvénient de ces deux techniques est la surgénération de structures. Le système est considérablement ralenti. En outre, les scores de sélection de la meilleure analyse, calculés sur des préférences psycholinguistiques pour des phrases grammaticalement correctes, ne fonctionnent pas. Il a donc fallu développer un algorithme d'identification de l'analyse préférentielle (Potter, 2002) basé (i) sur des heuristiques éliminant les analyses improbables, (ii) un classement sur la fréquence des mots et (iii) un classement sur la fréquence des types d'erreurs. En outre, la réinterprétation phonologique exige de telles ressources que nous l'avons limitée aux phrases relativement courtes et qui ne sont pas trop segmentées.

La version finale des outils de diagnostic n'a malheureusement pas pu être testée par des apprenants⁴ mais nous disposons toutefois d'une évaluation basée sur un corpus de 120 phrases représentatives des erreurs à détecter, qui avaient été récoltées lors de l'utilisation d'une version antérieure du logiciel. Des résultats plus détaillés sont donnés dans (Vandeventer Faltin, 2003, L'haire & Vandeventer Faltin, 2003a). Il en ressort que les performances sont comparables à celle du correcteur de Microsoft Word® 2000. Cependant, notre système pêche par une trop grande surdétection d'erreurs. Les scores de certaines erreurs comme l'oubli du tiret sont mauvais et le diagnostic de celles-ci pourrait être désactivé. Il serait sans doute utile d'introduire plusieurs niveaux d'erreurs et de leur donner un plus ou moins grand poids, afin de pondérer le diagnostic. Néanmoins, cette démarche nécessiterait la récolte et l'étude d'un corpus d'erreurs conséquent.

Une évaluation détaillée de la réinterprétation phonologique permettrait sans doute un affinage de cette technique, notamment pour éliminer certains choix absurdes et pour

⁴ Un test d'un premier prototype a eu lieu entre novembre 2001 et mars 2002 chez 13 enseignants de FLE de divers pays. Entre février et juin 2002, 150 étudiants de FLE belges et britanniques ont testé un produit corrigeant certaines erreurs d'accord. En janvier 2003, le produit quasi-final a été testé par des spécialistes d'ALAO et d'autres experts de divers pays, dont les observations ont permis de suggérer certaines des améliorations mentionnées à la section 5 de cet article.

proposer des mots qui ne sont pas des homophones au sens strict mais qui sont mal distingués par les apprenants (diphthongues). Enfin, il paraît indispensable de tenter de rétablir les accents qui sont souvent omis (participes passés) et de pallier l'absence d'apostrophes (**quand j etais petite*). De même, une série d'erreurs communes pourrait être détectées par *pattern matching* ou laissées aux techniques de correction orthographique : expressions idiomatiques de type *au lieu de*, confusion *du-dû*, etc. Les avertissements devraient pouvoir être déclenchés par le tuteur ou par l'apprenant une fois que ces notions sont considérées comme maîtrisées.

4. Interface du diagnostic

Après avoir décrit les techniques de diagnostic, nous passons maintenant aux interfaces proposées aux apprenants. Il est essentiel de fournir une rétroaction la plus précise et la plus utile possible (Levin & Evans, 1995 : 90), puisque l'apprenant ne peut se baser que sur l'ordinateur pour améliorer sa production.

L'analyseur FIPS produit un document XML qui sert de base à ses différentes applications (synthèse vocale, affichage etc.). Ce formalisme est suffisamment souple et évolutif tout en restant contraint. Il est également largement répandu et peut être traité par de nombreux logiciels et langages de programmation. Considérons la figure 1 qui contient un extrait de la sortie pour une phrase contenant une erreur d'accord.

```
<PROJ cat="TP" tree="Ph">
  <PROJ cat="DP" attachedAs="spec" colorGramm="sujet" tree="GN">
    <HEAD cat="D" tool="yes" gender="masc" number="sin" lexeme="le"
ortho="le" colorGrammLex="Det">
      <ERROR index="Aa01" manypart="yes" category="NBR">Le</ERROR></HEAD>
    <PROJ cat="NP" attachedAs="comp" tree="hidden">
      <HEAD cat="N" tool="no" gender="masc" number="plu" lexeme="chat"
ortho="chats" colorGrammLex="Nom">
        <PARTERROR antecedent="Aa01">chats</PARTERROR></HEAD>
      </PROJ></PROJ>

    <BAR cat="T" colorGramm="pred" tree="GV" index="AAAAA" number="sin"
person="3" tense="passeCompose" mode="indicatif" voice="active"
lexeme="dormir">
      <HEAD cat="V" tool="yes" coref="AAAAA" number="sin" person="3"
tense="present" mode="indicatif" lexeme="avoir" ortho="a"
colorGrammLex="Aux">a</HEAD>
      <PROJ cat="VP" attachedAs="comp" tree="hidden">

        <HEAD cat="V" tool="no" coref="AAAAA" gender="masc" number="sin"
tense="participePasse" lexeme="dormir" ortho="dormi"
colorGrammLex="Verbe">dormi</HEAD></PROJ></BAR></PROJ>
```

Figure 1: Extrait de sortie XML pour "le chats a dormi"

L'analyse contient des projections syntaxiques <PROJ>. A côté des catégories communément utilisées par FIPS, ces balises contiennent un attribut *colorGramm* pour les fonctions syntaxiques et une étiquette syntaxique simplifiée dans *tree*. Les têtes lexicales sont marquées par <HEAD> et contiennent des informations sur les traits des mots. La balise <BAR> contient les informations du niveau intermédiaire des projections lorsqu'elles sont utiles comme dans le cas présent. <ERROR> et <PARTERROR> représentent les deux parties d'une erreur. Elles contiennent notamment une catégorie d'erreur, ici une erreur de nombre. Si une erreur porte sur plusieurs mots, les balises sont coïncidées, car plusieurs erreurs

peuvent apparaître dans une même phrase. Ce même et unique document XML permet ensuite d'obtenir les trois sorties que nous présentons maintenant.

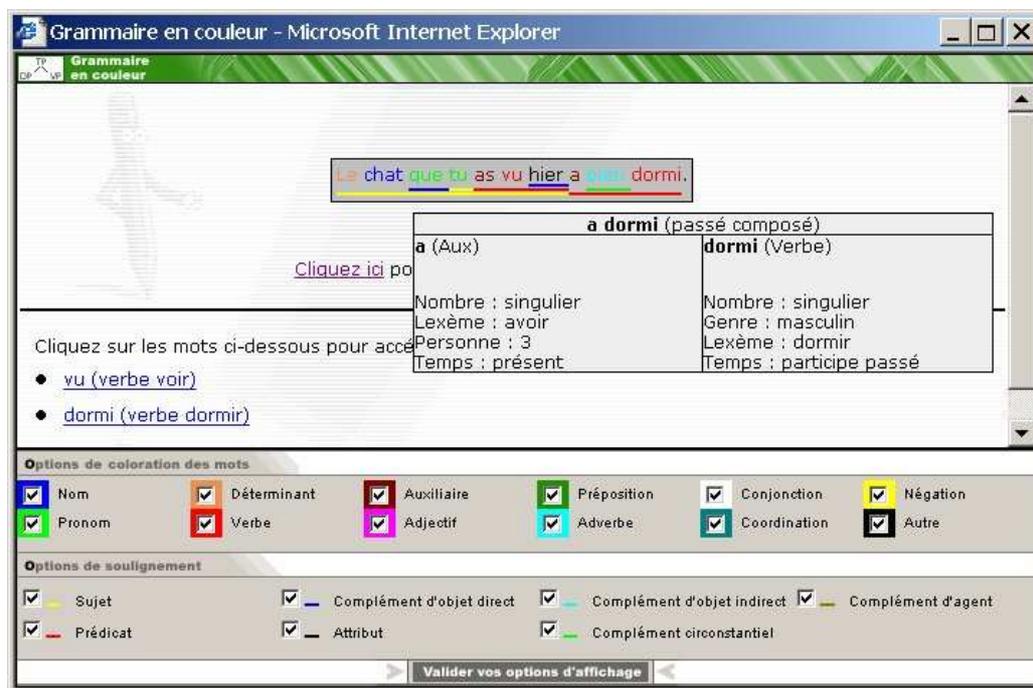


Figure 2: grammaire en couleur pour la phrase « Le chat que tu as vu hier a bien dormi. »

La figure 2 présente la sortie de la grammaire en couleurs. Les catégories lexicales sont représentées par des couleurs de police, les catégories syntaxiques par des soulignements. L'interface permet de sélectionner les fonctions à afficher. Un avertissement signale à l'apprenant si sa phrase contient des erreurs et un lien est disponible dans tous les cas vers le diagnostic. Si le curseur de la souris passe sur un mot, une bulle affiche les informations lexicales qui lui sont associées. Ici, nous regroupons les informations complètes sur le verbe au passé composé, en le décomposant en deux éléments, même lorsqu'un adverbe est entre l'auxiliaire et son participe. Les verbes de la phrase sont listés dans la fenêtre et un lien permet de lancer un conjugeur dynamique, qui les affiche à tous les temps et à tous les modes.

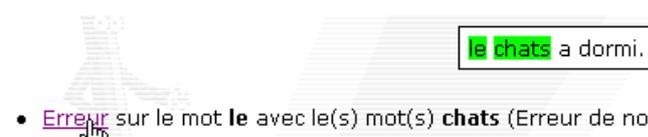


Figure 3: Système de diagnostic d'erreurs pour la phrase « *Le chats a dormi. »

La figure 3 présente le diagnostic d'erreur proprement dit. Les erreurs de la phrase sont énumérées avec les mots qu'elles comportent. Lorsque le curseur de la souris passe sur le mot *erreur*, les mots de l'erreur sont mis en surbrillance. En outre, les informations lexicales des mots sont affichées lorsque le curseur de la souris les survole. Enfin l'apprenant peut basculer vers la grammaire en couleurs. L'apprenant est donc en mesure de corriger lui-même sa production. Certaines erreurs sont considérées comme potentielles, comme les erreurs d'ordre de l'adverbe (*?Souvent il mange des glaces.*), qui sont particulièrement difficiles à traiter. Dans ce cas, le logiciel affiche l'erreur comme *warning*.

Enfin, nous proposons une analyse sous forme d'arbres syntaxiques, que nous ne présentons pas ici faute de place. Les arbres ne peuvent être affichés qu'en cas d'obtention d'une analyse complète de la phrase. Les étiquettes utilisées et la structure de la phrase sont simplifiées par rapport aux structures de FIPS. Des arcs relient certains nœuds de l'arbre, afin de montrer les relations syntaxiques pour les pronoms relatifs ou clitiques, pour le mouvement du passif, etc.

Les trois sorties que nous proposons fournissent un « feedback » suffisamment fiable, utile et complet aux apprenants. Malheureusement, aucun test n'a pu être mené auprès d'apprenants sur la version finale. Seul un prototype très incomplet a été testé au début du projet, qui ne détectait que quelques erreurs d'accord. Les réactions des apprenants étaient mitigées, ce qui nous permet d'espérer un meilleur accueil de la version complète.

5. Vers un feedback plus intelligent

Dans cette section, nous décrivons nos recherches actuelles et présentons des propositions d'utilisation intelligente de nos outils. Dans le cadre de **FreeText**, nous avons commencé à implémenter une technique de comparaison sémantique de phrases. En effet, la vérification syntaxique d'une phrase ne suffit pas pour vérifier la cohérence de la réponse. Examinons les phrases en (3).

(3)a. Qu'est-ce que Paul a fait ? b. Il a lu ses livres. c. Il a lu ces livres.

Supposons que la réponse attendue à la question (3a) soit (3b). L'apprenant peut répondre (3c) qui est tout à fait grammatical et correct en isolation. La confusion entre les deux pronoms homophones ces/ses est courante, même chez des locuteurs natifs. Une technique de *pattern matching* est possible pour filtrer des réponses, mais il est alors nécessaire de prévoir de nombreux cas de figure à l'avance et de maîtriser des formules compliquées. Une analyse sémantique peut donc être un apport intéressant. Dorr & al. (1995) proposent de comparer la phrase de l'apprenant à la réponse attendue stockée dans le système en se basant sur des *Lexical Conceptual Structures* (LCS, Jackendoff, 1990). Les LCS sont des représentations conceptuelles abstraites basées sur des primitives, utilisées par exemple dans le domaine de la traduction.

Nous proposons une démarche analogue qui utilise un autre formalisme de structure de traduction, les *Structures Pseudo-Sémantiques* (PSS, Etchegoyhen & Wehrle, 1998). Les PSS sont des structures hybrides qui combinent des informations abstraites sémantiques et aspectuelles à des informations lexicales (des mots concrets) pour les classes ouvertes (verbes, substantifs, adjectifs et adverbes). Elles se composent de trois sous-types : (i) clauses (verbes, adj.) ; (ii) syntagmes nominaux ; (iii) modifieurs (adj. et adv.). Ces structures peuvent se combiner entre elles dans une structure hiérarchique qui dénote les relations profondes des éléments entre eux.

Les PSS représentant la phrase *As-tu lu ce livre ?* ne différeront que par quelques traits des PSS de *Tu as lu ce livre*, de *Ce livre a été lu* ou de *C'est ce livre que tu as lu*. Cette proximité de structures est exploitable pour des exercices de transformation. De plus, les structures utilisées sont moins complexes et plus faciles à parcourir, à comparer et à manipuler que les analyses syntaxiques de FIPS. Les PSS et leur emploi pour la comparaison de phrases est décrite plus en détail chez L'haire & Vandeventer Faltin (2003a, b). Malheureusement, par manque de ressources, cette recherche n'a pas été implémentée totalement dans **FreeText**. Elle est considérée comme résultat secondaire du projet et fait l'objet de nos recherches

actuelles. Nous envisageons également l'emploi d'un dictionnaire sémantique pour permettre de détecter l'emploi de synonymes, hyponymes et hyperonymes. Il serait aussi possible d'utiliser les PSS pour un reformulateur de phrases, où l'apprenant pourrait modifier certaines valeurs et observer la nouvelle phrase produite par le système de génération.

Passons maintenant aux améliorations possibles des interfaces. Plusieurs modifications de nos sorties doivent être apportées. Certaines suggestions, basées sur une évaluation du logiciel par des professionnels, ont été faites par le consortium du projet à l'issue de celui-ci. Quelques considérations sont avant tout pratiques. En premier lieu, il serait préférable que plusieurs phrases puissent être traitées à la fois et que l'apprenant puisse revenir sur l'analyse de chacune des phrases. Au niveau de la grammaire en couleurs, les catégories lexicales seraient plus lisibles avec des fonds d'écran et des couleurs de caractères contrastives plutôt que par des simples couleurs de caractères sur des fonds uniformes.

Enfin, il n'apparaît pas clairement que l'analyseur n'a pas obtenu une analyse complète. Si un apprenant est conscient que l'analyse a échoué, il peut reformuler sa phrase en la simplifiant ou même ignorer la phrase et demander la réponse si elle est disponible ; mieux encore, les morceaux d'analyse et les informations lexicales affichées pourraient être une aide à l'autoremédiation. Pour aider l'apprenant dans cette tâche, nous suggérerons l'emploi d'un dictionnaire intégré au logiciel, doté d'un système de recherche phonétique adapté aux confusions fréquentes chez les apprenants de langues. L'apprenant pourrait alors développer des stratégies semblables à celles de la réinterprétation phonétique présentée à la section 3⁵. Développer de tels outils nécessiterait une typologie des erreurs qui n'est pas pour l'instant contenue dans FRIDA. En outre, la valeur pédagogique de tels outils devrait être testée spécifiquement.

Notre logiciel dispose d'une grammaire de référence qui a été finement indexée. La sortie XML dont nous disposons permettrait de renvoyer directement à la bonne section, à l'aide d'une simple table de correspondances entre le type d'erreur et la partie du discours d'une part et une section de la grammaire d'autre part. A l'inverse, il faudrait créer un lien entre certaines phrases d'exemple de la grammaire de référence, voire certaines phrases des textes des tutoriels, et la grammaire en couleurs et/ou les arbres, afin de sensibiliser l'apprenant à certains phénomènes grammaticaux. Dans ce cas, il faudrait n'afficher que les catégories lexicales et syntaxiques pertinentes, l'apprenant étant libre ensuite d'afficher une analyse complète.

Pour le diagnostic d'erreurs, pour faciliter l'autoremédiation, l'on devrait pouvoir modifier la phrase directement dans la fenêtre même du diagnostic. D'autres améliorations demandent plus de réflexion. Le feedback serait plus efficace s'il était affiché graduellement. En effet, un premier feedback pourrait signaler uniquement le(s) lieu(x) de l'erreur. Un second pourrait ajouter le type de l'erreur. Au troisième niveau, on afficherait tous les traits lexicaux, comme dans la sortie actuelle. Aux niveaux suivants, il serait envisageable d'offrir une analyse syntaxique de la phrase, d'offrir des solutions alternatives, parmi lesquelles figurerait la bonne réponse, voire même de corriger automatiquement l'erreur. Comme Heift (2003), nous devons

⁵ Par exemple, nous pourrions utiliser un outil comme le *Dictionnaire d'Apprentissage du Français Langue Etrangère ou Seconde* (DAFLES, Selva & al., 2003, DAFLES, 2003), qui présente de nombreuses caractéristiques utiles pour les apprenants. Ce dictionnaire ne permet cependant pas de recherche phonétique.

aussi envisager de ne présenter qu'une erreur à la fois afin de ne pas saturer l'apprenant d'informations.

Pour la sortie arborescente, nous générons actuellement une image statique qui est peu maniable, souvent trop large et trop détaillée. Un étudiant travaille actuellement à la réalisation d'une sortie où les informations lexicales seront affichées au passage de la souris et où certaines branches pourront être réduites⁶. Cette nouvelle forme sera plus lisible et devrait être plus profitable aux apprenants.

Enfin, tant l'apprenant que le tuteur devraient pouvoir paramétrer les erreurs à détecter. Les utilisateurs natifs de certains correcteurs grammaticaux commerciaux ont souvent désactivé ces outils à cause de leurs corrections stylistiques intempestives, par exemple contre l'usage de la voix passive. Un apprenant qui estime qu'il maîtrise une notion doit pouvoir désactiver certains avertissements. Enfin, les concepteurs d'exercices doivent pouvoir spécifier quelles erreurs doivent être ciblées et donc affichées par le diagnostic d'erreurs et choisir de masquer les autres erreurs dans un premier temps, voire même désactiver complètement la détection de certaines erreurs.

Pour terminer, évoquons des pistes à plus long terme. La sortie de nos outils convient très bien aux produits d'apprentissage *intelligemment* assisté par ordinateur (ALIAO, ICALL en anglais). Les logiciels dits intelligents modélisent le processus d'apprentissage ainsi que les connaissances des apprenants et proposent dynamiquement les documents, exercices et autres activités les plus appropriés aux besoins d'apprentissage. Notre sortie XML permet d'extraire de nombreux enseignements sur l'apprenant pour le module-élève : non seulement, il est possible de détecter les erreurs les plus fréquentes, mais aussi les structures syntaxiques et le vocabulaire qui sont maîtrisés et utilisés. Les lacunes des apprenants seraient détectées grâce à un prétest avant l'utilisation du logiciel. De même, une série de connaissances objectives telles que la langue maternelle ou usuelle de l'apprenant sont essentielles : un germanophone utilisera **aider à quelqu'un* au lieu d'*aider quelqu'un*, par surgénéralisation de l'allemand *helfen + datif*. Le corpus FRIDA pourrait donner des enseignements fort utiles pour la connaissance de ces profils. Au niveau phonologique, on préconisera un affinage de la correction orthographique voire grammaticale pour la confusion *l/r* des locuteurs asiatiques, *b/v* pour les hispaniques ou les diphtongues pour la plupart des locuteurs non natifs.

6. Conclusion

Par rapport à nos ambitions de départ, **FreeText** a vu ses ambitions réduites. La technique de la comparaison de phrases a dû être reportée après la fin du projet et les performances du système de diagnostic peuvent sembler relativement faibles par rapport aux espoirs soulevés. Il est aussi un peu frustrant de ne pas disposer de résultats sur l'accueil réel de nos outils d'aide chez les apprenants. Nous ne disposons que d'une évaluation de spécialistes et des réactions suscitées par les diverses présentations du logiciel que nous avons faites. Une analyse plus fine de nos résultats et des imperfections de nos techniques devrait améliorer ces résultats. L'exploitation *intelligente* de notre sortie pourrait aussi être d'un grand apport pédagogique. Nous en concluons qu'il serait tout à fait possible et souhaitable d'utiliser nos

⁶ Si un nœud est réduit, nous utiliserons la convention linguistique d'écrire tous les mots qui figurent sous le nœud réduit et de les surmonter d'un triangle.

outils moyennant quelques améliorations qui semblent tout à fait implémentables à court terme.

Références

Bibliographie

- Chomsky, N. & Lasnik, H. (1995). “The theory of Principles and Parameters”. In N. Chomsky (Ed). *The Minimalist Program* (pp. 13-127). Cambridge, Mass.: MIT Press.
- Dorr, B., Hendler, J., Blanksteen, S. & Migdaloff B. (1995). “On Beyond Syntax: Use of Lexical Conceptual Structure for Intelligent Tutoring”. In Holland, V. M., Kaplan, J. D & Sams, M. R., Eds., *Intelligent Language Tutors: Theory Shaping Technology*, p. 289–309. Mahwah, NJ: Lawrence Erlbaum Associates.
- Etchegoyhen T. & Wehrle T. (1998). “Overview of GBGen: A large-scale, domain independent syntactic generator”. In *Proceedings of the 9th International Workshop on Natural Language Generation* (pp. 288-291). Niagara-on-the-Lake: Association for Computational Linguistics.
- Granger, S. (2003). “Error-tagged learner corpora and CALL: a promising synergy”. In Heift, T. & Schulze, M. (Eds). *Error Analysis and Error Correction in Computer-Assisted Language Learning. CALICO*, vol. 20, n° 3. pp. 465-480.
- Heift, T. (2003). “Multiple Learner Errors and Feedback: A Challenge for ICALL Systems”. In Heift, T. & Schulze, M. Eds. (2003). *Error Analysis and Error Correction in Computer-Assisted Language Learning. CALICO*, vol. 20, n° 3. pp. 533-548.
- Jackendoff, R. S. (1990). *Semantic Structures*. Cambridge MA, USA: MITPress.
- Levin, L. S. & Evans, D. A. (1995). “ALICE-chan: A Case Study in ICALL Theory and Practice”. In Holland, V. M., Kaplan, J. D & Sams, M. R., Eds., *Intelligent Language Tutors: Theory Shaping Technology*, p. 77–97. Mahwah, NJ: Lawrence Erlbaum Associates.
- L’haire, S. & Vandeventer Faltin, A. (2003a). « Diagnostic d’erreurs dans le projet FreeText ». *Alsic*, Vol. 3, n° 2. pp. 21–37. Consulté en août 2004. http://alsic.u-strasbg.fr/Num11/lhaire/alsic_n11-rec3.htm
- L’haire, S. & Vandeventer Faltin, A. (2003b). “Error Diagnosis in the FreeText Project”. *CALICO*, vol. 20, n° 3, pp. 481–495.
- Potter, K. R. (2002, November 28). “Taming overgeneration in a constraint-relaxed parser”. Paper presented at the 13th meeting of Computational Linguistics in the Netherlands, Groningen.
- Selva, T., Verlinde, S. & Binon, J. (2003). Vers une deuxième génération de dictionnaires électroniques. *Traitement automatique des langues*, Vol. 44, n° 2, pp. 177–197.
- Tschichold, C. (2003). “Lexically Driven Error Detection and Correction”. *Calico Journal*, vol. 20, n° 3, pp. 549–539.
- Vandeventer Faltin, A. (2003). *Syntactic Error Diagnosis in the context of Computer Assisted Language Learning*. Thèse de Doctorat, Université de Genève, Faculté des Lettres, Ms..

Wehrli, E. (1997). *L'analyse syntaxique des langues naturelles: Problèmes et méthodes*. Paris: Masson.

Sites Internet

DAFLES (2003). Consulté le 30 juillet 2004. <http://www.kuleuven.ac.be/dafles/>

FreeText (2002). **FreeText** : French in Context: An advanced hypermedia CALL system featuring NLP tools for a smart treatment of authentic documents and free production exercises. Consulté le 16 août 2004. <http://www.latl.unige.ch/freetext/>

Logiciel

Hugo 2000 (2000). Softissimo SA: Paris.