Cet article a été publié dans les actes de la 18^{me} Conférence sur le Traitement Automatique des Langues Naturelles 2011. La publication originale est disponible sur le site `www.taln2011.org`

Le TAL au service de l'ALAO/ELAO
L'exemple des exercices de dictée automatisés

Richard Beaufort Sophie Roekhaut
CENTAL, UCLouvain, Place Blaise Pascal 1, B-1348 Louvain-la-Neuve
{richard.beaufort,sophie.roekhaut}@uclouvain.be

Résumé. Ce papier s'inscrit dans le cadre général de l'Apprentissage et de l'Enseignement des Langues Assistés par Ordinateur, et concerne plus particulièrement l'automatisation des exercices de dictée. Il présente une méthode de correction des copies d'apprenants qui se veut originale en deux points. Premièrement, la méthode exploite la composition d'automates à états finis pour détecter et pour analyser les erreurs. Deuxièmement, elle repose sur une analyse morphosyntaxique automatique de l'original de la dictée, ce qui facilite la production de diagnostics.

Abstract. This paper comes within the scope of the Computer Assisted Language Learning framework, and addresses more especially the automation of dictation exercises. It presents a correction method of learners' copies that is original in two ways. First, the method exploits the composition of finite-state automata, to both detect and analyze the errors. Second, it relies on an automatic morphosyntactic analysis of the original dictation, which makes it easier to produce diagnoses.

Mots-clés : ALAO/ELAO, exercices de dictée, alignement, diagnostic, machines à états finis.

Keywords: CALL, dictation exercises, alignment, diagnosis, finite-state machines.

1 Introduction

L'Apprentissage et l'Enseignement des Langues Assistés par Ordinateur (ALAO/ELAO) ont pour objectif premier d'améliorer l'acquisition des langues par les apprenants. Pourtant, force est de constater qu'actuellement, l'investissement dans le domaine a plus été technologique que didactique : pour l'essentiel, la numérisation des cours n'a pas modifié leur contenu ni leurs méthodes d'évaluation (Desmet & Héroguel, 2005). Selon les spécialistes, l'amélioration de l'apprentissage et de l'enseignement implique de dépasser les sempiternels exercices fermés, tels que les textes à trous et les choix multiples qui, s'ils sont faciles à corriger, limitent considérablement les possibilités d'évaluation des connaissances. Il faudrait au moins proposer des exercices semi-ouverts, qui autorisent plusieurs réponses relativement prévisibles, pour autant que la correction automatique de ces exercices soit fiable : apprentissage et enseignement, en effet, ne tolèrent pas l'approximation (Antoniadis *et al.*, 2009).
La dictée, où l'enseignant lit à haute voix un passage que les étudiants doivent copier, est typiquement l'un de ces exercices semi-ouverts qui, s'il était automatisé, pourrait considérablement améliorer l'apprentissage et l'enseignement des langues. La dictée est ainsi un très bon moyen d'estimer le niveau d'un étudiant (Coniam, 1996). Sa pratique, en outre, permet d'améliorer des compétences telles que la maîtrise de la grammaire, les capacités de lecture, la connaissance du vocabulaire et le niveau de compréhension (Rahimi, 2008). Actuellement pourtant, rares ont été les essais d'automatisation de cet exercice. Or, grâce aux synthétiseurs de la parole, lire automatiquement un texte inconnu n'est plus un problème. Mais la correction d'une copie d'apprenant, par contre, est une étape beaucoup plus délicate à automatiser, parce qu'elle pose deux questions sensibles : la détection de la place réelle des erreurs et leur classification.

2 État de l'art

L'importance d'exercices semi-ouverts en ALAO/ELAO a souvent été mise en avant. Des tests ont en outre montré l'intérêt de la dictée comme moyen d'évaluation et d'amélioration de la maîtrise de la langue, qu'il s'agisse d'apprenants natifs ou d'allophones, pour autant que ceux-ci présentent un niveau de maîtrise avancé de la langue étudiée (niveaux C1 ou C2 du *Cadre européen commun de référence pour les langues*). Pourtant, peu de travaux ont directement concerné l'automatisation de cet exercice et de sa correction.
Le travail le plus significatif dans le domaine est certainement celui de Santiago-Oriola (1998). La partie de ce travail qui nous intéresse ici est la méthode de correction proposée : elle repose sur le fait que le lien entre graphie et prononciation n'est pas aisé à acquérir en français : seuls 80 à 85% des lettres d'un texte transcrivent un phonème, ce qui est la cause de nombreuses fautes d'orthographe. Sur cette base, la correction proposée se divise en deux modules : un premier module s'occupe de la détection des erreurs en réalisant un alignement de l'original et de la copie dirigé par des règles de transformation phonologiques ; un second module ne produit pas un diagnostic, mais *sélectionne* un diagnostic *pré-établi*, associé dans les ressources du système à une ou plusieurs erreurs recensées dans la langue. Hormis une évaluation de l'outil qui a montré que 80% des 24 enfants l'ayant testé l'ont apprécié, ce travail très intéressant n'a malheureusement pas été poursuivi.
Plus récemment, le logiciel de dictées *La Dictée interactive* a été présenté dans la revue *Alsic*, dédiée à l'apprentissage des langues (Ruggia, 2000). Cet article, centré sur la présentation du potentiel de l'outil, ne donne qu'une seule information concernant la méthode de correction des erreurs : elle cible les erreurs courantes chez les apprenants italophones de niveau faux-débutant en français. Cette méthode est donc probablement peu générique.
Ces dernières années, la dictée a complètement disparu de la littérature scientifique. Dans le domaine des exercices semi-ouverts, on peut cependant encore noter les travaux de Desmet & Héroguel (2005), dont la plateforme d'apprentissage des langues étrangères autorise entre autres la réalisation de traductions de phrases d'une langue source vers une langue cible. Le principe de correction proposé se rapproche de l'exercice de dictée dans lequel l'original est disponible : l'idée, ici, est de produire plusieurs formulations (les « originaux »), et de sélectionner, par *approximate string matching*, la formulation dont la réponse de l'apprenant se rapproche le plus. Le système, qui travaille au niveau du mot, signale ensuite les erreurs à l'apprenant en remplaçant un mauvais mot par XXX, un mot superflu par (XXX) et un mot manquant, par (...). Par contre, aucun diagnostic n'est produit.

3 Le système de correction proposé

Notre méthode de correction partage deux similarités avec celle de Santiago-Oriola (1998). Premièrement, elle comprend deux étapes, une phase de détection précédant une phase de diagnostic. Deuxièmement, la phase de détection repose sur un alignement de l'original et de la copie.
Un aspect distingue particulièrement notre approche : la totalité de la correction repose sur une analyse morphosyntaxique de l'original de la dictée. Cette analyse est produite par le système eLite (Beaufort & Ruelle, 2006), qui réalise successivement un pré-traitement, une analyse morphologique et une désambiguïsation contextuelle du texte. La désambiguïsation est obtenue par l'application d'un modèle de langue probabiliste (Beaufort *et al.*, 2002). Les informations morphosyntaxiques sont ensuite stockées dans une structure de données qui contient, entre autres, une couche *Sent* correspondant aux phrases, et une couche *Word* correspondant aux mots.
Une fois l'analyse de l'original d'une dictée terminée, la structure de données correspondante est sauvegardée dans un fichier XML, qui sera chargé par notre module de correction lorsqu'il devra traiter une copie de cette dictée. Au-delà des informations morphosyntaxiques qui seront utilisées par le module de correction, il faut noter que le processus de correction dans son ensemble est influencé par cette structure de données : la couche *Sent*, qui identifie les phrases du texte, permet en effet d'appliquer le module de correction phrase par phrase, ce qui réduit considérablement la complexité du processus.

Détection. Comme dans l'état de l'art, la détection des erreurs de l'apprenant se fait sur la base d'un alignement de la phrase originale et de la copie. Classiquement, cet alignement est obtenu en calculant la distance d'édition des deux séquences (Damerau, 1964; Levenshtein, 1966). Or, la distance d'édition classique n'autorise que des opérations de base : substitution, insertion et suppression d'un caractère, et transposition de deux caractères contigus. Ceci est gênant dans le cas qui nous occupe, parce que les erreurs des apprenants correspondent souvent à des substitutions n-m, où n caractères se substituent à m caractères : *-es* $\leftrightarrow$ *-ent*, *-ait* $\leftrightarrow$ *-aient*, *-er* $\leftrightarrow$ *-ées*, etc. Dans

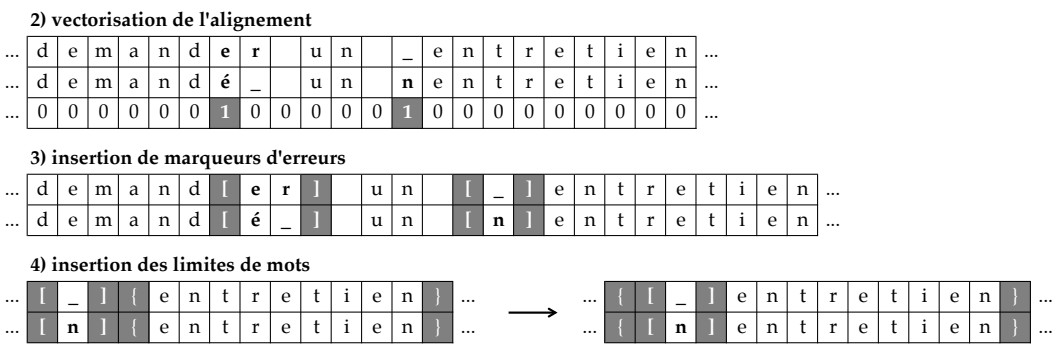


FIGURE 1 – Illustration des étapes 2 à 4 de la détection des erreurs

le cadre de la distance d'édition classique, la substitution n-m se modélise sous la forme de plusieurs opérations d'édition, ce qui tend à l'écarter des solutions pertinentes, étant donné que la distance qui lui est attribuée est la somme de plusieurs opérations. Afin de dépasser cette limite, nous avons eu recours aux machines à états finis et à une méthode que nous avons décrite dans (Beaufort, 2010) : étant donné deux séquences x et y représentées sous la forme des automates à états finis $\mathcal{X}$ et $\mathcal{Y}$, nous construisons le transducteur pondéré $\mathcal{E}$ correspondant à l'ensemble E des alignements possibles entre x pour y. Cet ensemble est obtenu au travers de la cascade de compositions :

$$\mathcal{E} = \mathcal{X} \circ \mathcal{F} \circ \mathcal{Y} \tag{1}$$

où $\mathcal{F}$ est un transducteur pondéré qui modélise les opérations d'édition acceptées. Le meilleur alignement entre x et y correspond au meilleur chemin de $\mathcal{E}$, obtenu par calcul du plus court chemin d'un graphe. La méthode est appelée *composition filtrée*, parce que le transducteur pondéré $\mathcal{F}$ peut être considéré comme un filtre qui détermine la taille de l'intersection entre x et y. Le filtre F est compilé sous la forme d'un transducteur $\mathcal{F}$ à partir d'un fichier de règles de réécriture de la forme :

$$\phi \quad ?\rightarrow \quad \psi \quad / \quad w \tag{2}$$

où la séquence ϕ *peut* se réécrire ψ et se voit dans ce cas attribuer le poids w. Il s'agit donc de règles *facultatives*, ce qui permet à ϕ soit de rester inchangé, soit de se voir appliquer plusieurs réécritures.

L'algorithme de détection des erreurs, illustré en Figure 1, est entièrement construit autour de ce principe :

1. la phrase originale x et la copie y sont converties en automates à états finis $\mathcal{X}$ et $\mathcal{Y}$. Les deux automates sont composés au travers du filtre $\mathcal{F}$ qui représente les opérations d'édition classiques ainsi que la substitution n-m des séquences graphiques couramment confondues en français. Actuellement, le filtre autorise surtout la substitution des terminaisons nominales et verbales, telles que celles que nous avons citées précédemment. Le transducteur $\mathcal{E}$ des alignements possibles entre x et y est ensuite réduit à son meilleur chemin $\mathcal{E}'$, correspondant au meilleur alignement de x et y ;
2. le transducteur $\mathcal{E}'$ est parcouru et converti en trois vecteurs : un pour la phrase originale, un pour la copie, et un pour les poids associés aux opérations réalisées. Dans le vecteur de poids, un poids positif indique le début d'une opération d'édition ;
3. les trois vecteurs sont parcourus en parallèle afin d'entourer les erreurs de marqueurs [et] qui en indiquent les limites. Le système considère qu'une erreur commence quand le poids est différent de 0, et qu'elle finit lorsque le poids redevient 0 et que les deux vecteurs de lettres présentent des caractères identiques ;
4. il reste à identifier les frontières de mots à l'aide de marqueurs { et }. À ce niveau, l'algorithme est guidé par l'analyse linguistique, qui parcourt successivement les éléments de la couche *Word*. Sur cette base, le système identifie le mot courant dans le vecteur de la phrase originale. Ensuite, il adapte les frontières au besoin, pour inclure dans le mot courant les erreurs contiguës qui correspondent à des insertions de caractères alphabétiques. C'est le cas du *n* dans la forme *nentretien* ;
5. chaque erreur est ensuite sauvegardée dans le *Word* adéquat de la structure de données. Cette sauvegarde s'accompagne de calculs permettant de catégoriser l'erreur : *dans ou en frontière d'élément*, *élément manquant*, *élément superflu*, *erreur ne contenant que des séparateurs*. Ces indices guideront l'établissement du diagnostic.

Diagnostic. À ce stade, toutes les informations disponibles sont sauvegardées dans les éléments *Word* de la structure de données : l'analyse morphosyntaxique de la forme correcte et, s'il y a eu erreur, la forme erronée et les indices prélevés. Le diagnostic n'est déclenché que sur les éléments *Word* contenant une erreur. Dans l'ensemble, les erreurs concernent (1) un séparateur, (2) un mot simple ou (3) une séquence d'éléments (mots et séparateurs). L'orientation du diagnostic vers une erreur sur séquence dépend de l'*indice* associé à l'erreur : un séparateur manquant peut indiquer une fusion en une forme du lexique (*quoi que* → *quoique*, *d'avantage* → *davantage*), un séparateur superflu, une segmentation en plusieurs formes du lexique (*quoique* → *quoi que*, *davantage* → *d'avantage*). Si les indices l'y poussent, l'algorithme de diagnostic commence donc par tester une erreur sur séquence, et ne propose un autre diagnostic que si ces tests échouent. Pour des raisons de clarté cependant, nous commençons par détailler le fonctionnement du diagnostic sur séparateur et sur mot simple.

1) Le diagnostic relatif à une erreur sur séparateur (ponctuation ou espace) est très simple à produire : si l'indice sauvegardé est *superflu* ou *manquant*, le diagnostic est identique. Sinon, le séparateur existe mais est erroné, et le diagnostic signale simplement que l'on attendait un séparateur différent.

2) Une erreur sur mot simple peut être lexicale et/ou grammaticale. Une erreur est lexicale si la forme erronée est hors-vocabulaire (*nentretien*) ou appartient à la même catégorie que la forme correcte, mais possède un lemme différent (*sceptique* ↔ *septique*). Une erreur est grammaticale si la forme erronée présente des traits grammaticaux différents de ceux de la forme correcte (*parle* ↔ *parles* diffèrent au niveau de la personne). Une forme erronée peut bien sûr cumuler erreur lexicale et grammaticale (*différent* ↔ *différant* cumulent erreur de lemme et erreur de catégorie). Pour poser l'un de ces diagnostics, nous commençons par rechercher la forme erronée dans le lexique. Si la forme n'y figure pas, elle est considérée comme hors-vocabulaire. Dans le cas contraire, l'idée est de comparer l'analyse linguistique retenue pour la forme correcte lors de la préparation de la dictée, à l'ensemble d'analyses possibles proposées par le lexique pour la forme erronée, et de retenir l'analyse de la forme erronée *la plus proche* de celle de la forme correcte.
Qu'il s'agisse d'une forme correcte ou erronée, une analyse linguistique est toujours constituée d'un lemme et des traits grammaticaux suivants : temps/mode, genre, nombre, personne.
La méthode de comparaison d'analyse que nous utilisons est fort proche de la méthode d'alignement que nous avons présentée précédemment. Elle est illustrée en Figure 2 : on compile l'analyse de la forme correcte (a_1) et l'ensemble d'analyses de la forme erronées (a_2) sous la forme d'automates à états finis (resp. $\mathcal{A}_1$ et $\mathcal{A}_2$). Sur cette base, la meilleure analyse à conserver pour la forme erronée ($\mathcal{A}'_2$) correspond au meilleur chemin de la composition de ces deux automates au travers d'un filtre $\mathcal{F}_t$:

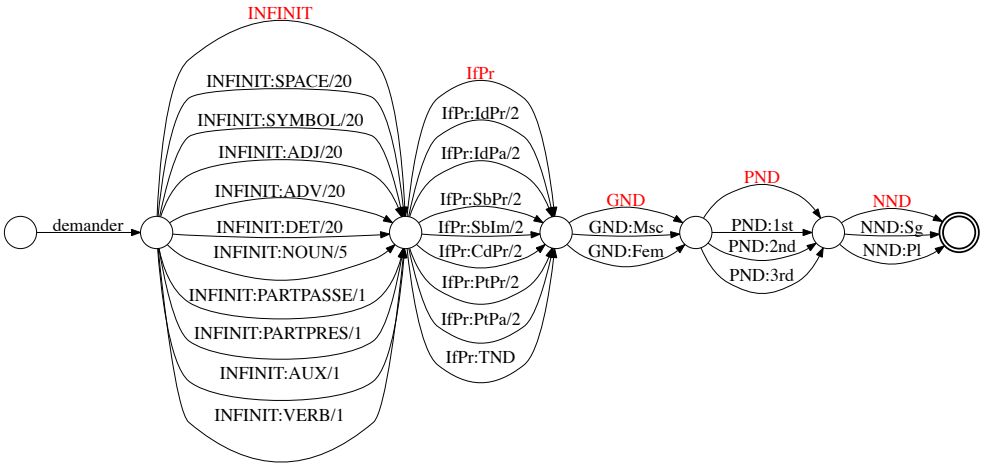
$$\mathcal{A}'_2 = \text{Best}(\mathcal{A}_1 \circ \mathcal{F}_t \circ \mathcal{A}_2) \tag{3}$$

où le filtre autorise des conversions pondérées entre traits grammaticaux. Par exemple, un infinitif peut être converti en participe passé moyennant un coût de 1 et en nom moyennant un coût de 5. Le meilleur chemin entre les deux analyses est donc celui qui réalise les transformations de traits les moins coûteuses.
Lorsque les lemmes des deux formes diffèrent (*sceptique* ↔ *septique*), la composition des automates échoue. Dans ce cas, l'erreur est au moins lexicale. Il reste cependant à choisir l'analyse de la forme erronée et à tester une éventuelle erreur grammaticale. Le même calcul est de ce fait reproduit sur deux nouveaux automates, ne présentant que les traits grammaticaux des deux formes à comparer. Cette composition donne toujours un résultat.

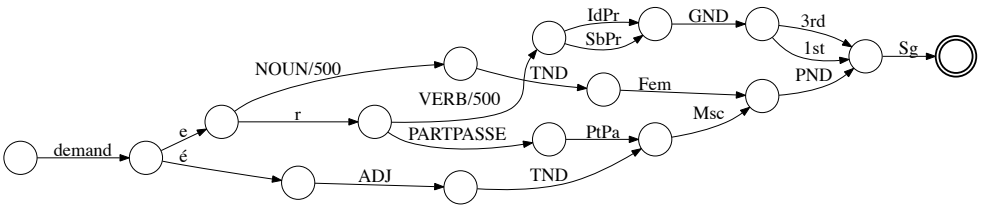
3) Dans le principe, l'analyse d'une erreur sur séquence se déroule comme celle d'un mot simple : la séquence erronée est recherchée dans un lexique. S'il n'y a pas de résultat cependant, la séquence n'est pas considérée comme hors-vocabulaire ; le diagnostic s'oriente simplement vers l'un des deux autres types d'erreurs.
La recherche de la séquence erronée dans le lexique diffère en deux points de la recherche d'un mot simple : il faut *construire* la séquence à rechercher et *choisir* un lexique approprié.
En cas de séparateur manquant (*quoique* pour *quoi que*, *davantage* pour *d'avantage*), on suppose que la forme erronée est un mot simple. Les mots corrects (par exemple, *quoi* et *que*) sont dans ce cas concaténés sans séparateur (*quoique*) et recherchés dans le même lexique que celui utilisé pour l'analyse des erreurs sur mots simples.
En cas de séparateur superflu (*quoi que* pour *quoique*, *d'avantage* pour *davantage*), on suppose que la forme erronée contient plusieurs formes correctes simples. Les segments de mots (par exemple, *d* et *avantage*) sont dans ce cas concaténés autour du séparateur superflu (*d'avantage*) et recherchés dans un lexique correspondant à l'expression régulière suivante :

$$(WordApo \quad | \quad (Word \quad Sep))^+ \quad Word \tag{4}$$

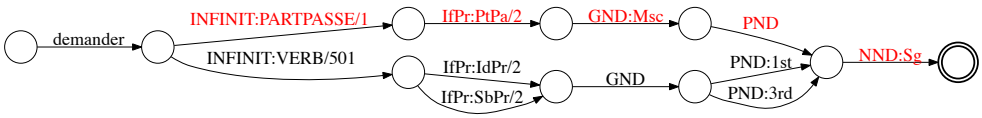
où $WordApo$ est un mot terminé par une apostrophe (*d'*, *qu'*, etc.) et Sep est un espace ou un trait d'union. Cette expression autorise donc simplement une suite de mots respectant les conventions typographiques du français.



1. Automate $\mathcal{A}_1$ représentant l'analyse de la forme correcte.
Ici, l'analyse est augmentée par composition avec le filtre $\mathcal{F}_t$



2. Automate $\mathcal{A}_2$ représentant l'ensemble des analyses de la forme erronée



3. Intersection des deux automates. Le chemin rouge correspond à la meilleure analyse

FIGURE 2 – Aide au diagnostic : composition des analyses

4 Première évaluation et travaux futurs

Nous avons réalisé une première évaluation sur le corpus de dictées de *Lenoble-Pinson*, numérisé et étiqueté par Fairon & Simon (2009). Ce corpus comprend 1 300 dictées réalisées sur une période de 40 années (1969–2008) aux Facultés Universitaires Saint-Louis (Bruxelles, Belgique) par des étudiants de 1^{re} et 2^{e} baccalauréat. Dans le corpus original, les erreurs étaient réparties en trois catégories : *usage*, *grammaire* et *ponctuation*. La version numérisée raffine ce classement en ajoutant trois catégories : *homophone*, *nouvelle orthographe* et *transcription*. Les fautes répertoriées *nouvelle orthographe* signifient que l'étudiant a signalé employer une convention (l'orthographe traditionnelle ou la nouvelle orthographe), mais a malgré tout utilisé des formes de l'autre convention. Les erreurs de transcription correspondent à des passages (un ou plusieurs mots complets) manquants ou ajoutés par l'étudiant. Il faut noter que le classement dans les différentes catégories est probablement à uniformiser : par exemple, la confusion *quelquefois* → *quelle que fois* est classée dans *usage*, alors que *quelque* ↔ *quel que* est classée dans *homophone*. Notre évaluation, réalisée sur un sous-ensemble de 441 dictées présentant 5 152 erreurs, a porté sur la qualité du système de détection d'une part, et sur la qualité du système de diagnostic d'autre part.

1) Globalement, 99% des erreurs (5 111 sur 5 152) sont correctement détectées et alignées. Toutes les erreurs de détection sont dues à des problèmes de transcription, qu'il s'agisse de passages superflus ou manquants. Le système est alors peu performant, puisqu'il ne gère correctement que 59% de ces erreurs. Pour comprendre le phénomène, voici un exemple de passage tronqué :

. . . française [, quels qu'en puissent être la gravité et le nombre] .

Contre toute attente, le système a favorisé l'alignement du *e* final de *française* sur le *ent* de *puissent*. Après analyse des poids des différents alignements possibles, cette erreur est en fait due au filtre, à qui l'on demande de favoriser la substitution *ent* ↔ *e*. Ce point ne manquera pas d'être étudié.

2) Évaluation du diagnostic. L'évaluation a porté ici sur l'ensemble des erreurs, sauf celles de transcription, dont la détection-même pose problème. Le corpus de test est de ce fait réduit à 5 052 erreurs. Nous ne générons pas le même classement que celui de Fairon & Simon (2009). L'évaluation a de ce fait consisté à valider manuellement les analyses produites, et ce à deux niveaux : en surface (séparateur superflu ou manquant, erreur lexicale et/ou grammaticale, etc.) et en profondeur (mot hors-vocabulaire, erreur de lemme, erreur sur un trait grammatical, etc.). Toute erreur d'analyse, quel que soit son niveau, a donné lieu au rejet du diagnostic dans son ensemble.
En l'état actuel, le système a été capable d'analyser correctement 83% des erreurs (4 200 sur 5 052). Ce résultat est à nuancer. 51,4% des erreurs d'analyse se répartissent entre les catégories *nouvelle orthographe*, *usage* et *homophone*. Actuellement, nous ne gérons pas la nouvelle orthographe, ce qui explique les erreurs dans cette catégorie : toute orthographe nouvelle, absente de nos lexiques, a été à tort classée comme hors-vocabulaire. Les erreurs recensées dans les catégories *usage* et *homophone* concernent toutes des séquences sur-segmentées, comme *quel que* pour *quelque* ou *d'avantage* pour *davantage*. Ceci est dû au fait que le système, au moment de l'évaluation, n'utilisait pas encore le lexique multi-mot décrit à l'Équation 4. Dans l'ensemble, ces erreurs devraient donc être facilement corrigées.
41,4% des erreurs concernent l'analyse grammaticale. L'erreur, dans ce cas, est souvent déjà présente dans l'analyse de la forme correcte. Il s'agit soit de confusions de catégories entre *nom* et *adjectif*, soit de confusions de modes entre *indicatif*, *subjonctif* et *impératif*, soit de confusions de personnes au singulier. Ce type d'erreurs indique clairement que l'analyse syntaxique réalisée sur la dictée doit être améliorée. Actuellement, nous pensons aborder le problème à l'aide de grammaires locales, qui guideront le modèle de langue.

Références

ANTONIADIS G., GRANGER S., KRAIF O., PONTON C. & ZAMPA V. (2009). NLP and CALL : integration is working. In *Proceedings of TaLC7, 7th Conference of Teaching and Language Corpora*.

BEAUFORT R. (2010). Composition filtrée et marqueurs de règles de réécriture pour une distance d'édition flexible. Application à la correction des mots hors-vocabulaire. *Traitement Automatique des Langues (T.A.L.)*, **51**(1), 11–40.

BEAUFORT R., DUTOIT T. & PAGEL V. (2002). Analyse syntaxique du français. Pondération par trigrammes lissés et classes d'ambiguïtés lexicales. In *Proc. JEP*, p. 133–136.

BEAUFORT R. & RUELLE A. (2006). eLite : Système de synthèse de la parole à orientation linguistique. In *Proc. JEP*, p. 509–512.

CONIAM D. (1996). Computerized dictation for assessing listening proficiency. *Calico Journal*, **13**, 73–86.

DAMERAU F. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, **7**(3), 171–176.

DESMET P. & HÉROGUEL A. (2005). Les enjeux de la création d'un environnement d'apprentissage électronique axé sur la compréhension orale à l'aide du système auteur IDIOMA-TIC. *Alsic. Apprentissage des Langues et Systèmes d'Information et de Communication*, **8**(1), 281–303.

FAIRON C. & SIMON A. (2009). Informatisation d'un corpus de dictées : 40 années de pratique orthographique (1967-2008). In *Pour l'amour des mots. Glanures lexicales, dictionnairiques, grammaticales et syntaxiques. Hommage à Michèle Lenoble-Pinson.*, p. 131–154.

LEVENSHTEIN V. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics*, **10**, 707–710.

RAHIMI M. (2008). Using Dictation to Improve Language Proficiency. *Asian EFL Journal*.

RUGGIA S. (2000). La dictée interactive. *Alsic. Apprentissage des Langues et Systèmes d'Information et de Communication*, **3**(1), 99–108.

SANTIAGO-ORIOLA C. (1998). *Système vocal interactif pour l'apprentissage des langues - la synthèse de la parole au service de la dictée.* PhD thesis, Toulouse III.