



**HAL**  
open science

## L'apport du TAL dans des environnements favorisant l'apprentissage auto-régulé

Mihai Dascalu, Philippe Dessus, Maryse Bianco, Mathieu Loiseau, Stefan  
Trausan-Matu

► **To cite this version:**

Mihai Dascalu, Philippe Dessus, Maryse Bianco, Mathieu Loiseau, Stefan Trausan-Matu. L'apport du TAL dans des environnements favorisant l'apprentissage auto-régulé. Journée EIAH&IA 2013, May 2013, Toulouse, France. pp.1. hal-00824285

**HAL Id: hal-00824285**

**<https://hal.science/hal-00824285>**

Submitted on 21 May 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# L'apport du TAL dans des environnements favorisant l'apprentissage auto-régulé

Mihai Dascalu<sup>1,2</sup>, Philippe Dessus<sup>1,2</sup>, Maryse Bianco<sup>2</sup>, Mathieu Loiseau<sup>3</sup>,  
et Stefan Trausan-Matu<sup>2</sup>

1 Univ. Grenoble Alpes, LSE, F-38040 Grenoble, France

{philippe.dessus, maryse.bianco}@upmf-grenoble.fr

2 Université Polytechnique de Bucarest, Roumanie

{mihai.dascalu, stefan.trausan}@cs.pub.ro

3 LIRIS, UMR 5205 CNRS, Universités Lyon-1 et Lyon-2, France

mathieu.loiseau@liris.cnrs.fr

**Résumé.** *Pensum* et *ReaderBench*, deux environnements personnels d'apprentissage dont le but est de favoriser l'apprentissage auto-régulé, sont tour à tour présentés. Le premier évalue automatiquement, dans les résumés ou synthèses de cours produites, les sauts de cohérence, les hors-sujets, et la pertinence des phrases. Le deuxième évalue la complexité textuelle des textes soumis à la lecture et les stratégies de lecture à partir des auto-explications des élèves. Leur architecture et les méthodes de traitement automatique de la langue utilisées sont décrites et discutées.

**Mots-clés.** Environnements Personnels d'Apprentissage, TAL, production écrite, lecture, complexité textuelle, Latent Semantic Analysis

## 1 Introduction

Les Environnements Personnels d'Apprentissage (EPA) sont des agrégats de petits programmes (*e.g.*, *widgets*), librement choisis, agencés et utilisés par l'utilisateur apprenant, lui permettant l'accès, l'organisation et la construction de matériel d'apprentissage [1, 2]. Ils insèrent l'apprenant dans un flux de travail plus aisément configurable et changeable qu'avec des outils standard (*e.g.*, systèmes de gestion de contenu). Ces programmes promeuvent souvent des aspects sociaux (web 2.0), mais plus largement, nous posons que tout système proposant de multiples activités librement sélectionnables est un EPA. Ils sont souvent vus comme plus à même que d'autres types de systèmes de favoriser un apprentissage auto-régulé, l'apprenant pouvant aisément se fixer des buts d'apprentissage ainsi que sa propre stratégie pour y parvenir. Des rétroactions peuvent également l'aider à réguler et évaluer son activité [3].

Nous avons conçu et implémenté deux EPA, *Pensum* [4] et *ReaderBench* [5, 6], partageant certaines caractéristiques : ils utilisent des techniques d'IA (comme LSA, *Latent Semantic Analysis* ou analyse sémantique latente, [7]) ou de TAL (comme

l'étiquetage grammatical) ; ils proposent des activités de lecture et d'écriture plutôt génériques, pouvant s'intégrer à de multiples activités d'apprentissage ; enfin, leurs rétroactions sont générées automatiquement. *ReaderBench* peut être vu comme la généralisation et l'enrichissement de certaines fonctionnalités de *Pensum*—*ReaderBench* inclut également des fonctionnalités d'analyse de forums et *chats*, qui ne sont pas décrites ici [6]. Ces outils sont conçus pour être utilisés dans un contexte d'enseignement à distance, dans lequel l'apprentissage auto-régulé n'est pas si souvent promu [8], même si une utilisation en enseignement en présence est bien sûr possible.

La section suivante de l'article présente la notion d'apprentissage auto-régulé, puis *Pensum* et *ReaderBench* seront successivement présentés, ainsi que leur architecture. Enfin, nous engagerons une discussion sur les méthodes d'IA/TAL utilisées.

## 2 Des EPA pour l'apprentissage auto-régulé

L'apprentissage auto-régulé représente l'apprenant et son activité à deux niveaux principaux [8]. Le premier niveau, celui de l'*objet*, concerne la manière dont l'apprenant va définir la tâche en cours, ses buts, et la stratégie nécessaire, pour mettre en œuvre diverses activités. Le deuxième niveau (*méta*), concerne la manière dont ces éléments vont être évalués au regard du contexte et des performances, qui, en retour, vont avoir un impact sur la définition des tâches, buts, stratégies et activités.

Nous nous intéressons aux deux activités les plus couramment mises en œuvre dans l'apprentissage auto-régulé, la lecture et l'écriture. *Pensum* et *ReaderBench* sont conçus pour engager l'apprenant dans des boucles d'activités lui permettant, le plus librement possible, de comprendre du matériel textuel suite à leur lecture et d'en (re)formuler l'essentiel (pour *Pensum*) ou d'explicitier ce qui a été compris (pour *ReaderBench*), par l'écriture d'un texte (*e.g.*, synthèse, résumé, explication).

Le **Tableau 1** ci-dessous décrit, pour chaque EPA, les boucles d'activités (colonne) et les activités cognitives engagées par l'apprenant et aidées par l'EPA *via* des rétroactions automatiques. Comprendre du matériel textuel *via* la lecture est sans doute l'activité la plus communément entreprise dans tout contexte d'enseignement/apprentissage, et se décline, dans l'activité d'écriture, par la production de résumés et synthèses de ce qui a été lu (pour *Pensum*), et par la verbalisation de ce qui a été compris (pour *ReaderBench*). Cette activité réside au niveau de l'*objet*, alors que l'activité d'(auto-)évaluation se situe au niveau *méta* et inclut, au niveau de l'activité de lecture, des fonctionnalités d'analyse automatique de la cohésion (pour *Pensum* et *ReaderBench*) et de la complexité textuelle (pour *ReaderBench*). Au niveau de l'activité d'écriture, différents paramètres liés à la cohésion textuelle sont analysés (*e.g.*, sauts de cohésion importants) les verbalisations de compréhension sont automatiquement analysées et les stratégies de lecture mises en œuvre sont catégorisées, selon la classification de Nardy et ses collègues [9], adaptée aux verbalisations enfantines à partir de McNamara [10].

Cette dernière activité d'(auto-)évaluation est à la fois réalisée par l'apprenant et par les EPA. Par exemple, *Pensum* détecte les sauts de cohésion trop importants entre

deux phrases, les affiche, mais l'apprenant peut signaler que ces sauts sont acceptables (contredisant ainsi le système). Un curseur permet de plus de régler le niveau de sévérité des évaluations.

**Tableau 1.** Activités principales envisagées dans *Pensum* et *ReaderBench*.

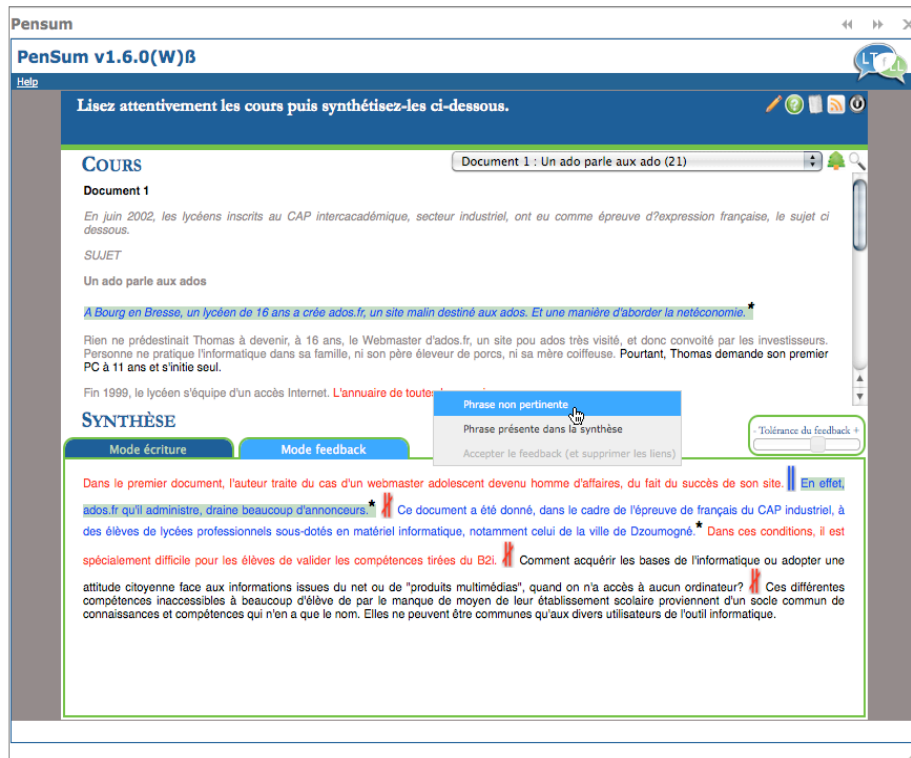
	Lecture	Ecriture
Comprendre	Matériel textuel : cours ( <i>Pensum</i> ), tout type de texte ( <i>ReaderBench</i> )	Production de résumés-synthèses ( <i>Pensum</i> ), auto-explication de compréhension ( <i>ReaderBench</i> )
(s'auto-)Evaluer	Cohésion textuelle ( <i>Pensum</i> et <i>ReaderBench</i> ), Complexité textuelle ( <i>ReaderBench</i> )	Saut de cohérence, hors-sujet et pertinence ( <i>Pensum</i> ), Stratégies de lecture ( <i>ReaderBench</i> )

### 3 *Pensum*, un EPA pour les synthèses de cours

*Pensum* est un outil d'aide à la synthèse de document(s), issu du projet européen *Learning Technologies for Lifelong Learning* (LtfLL, 7<sup>e</sup> PCRD-STREP). Toutefois sa genèse commence en amont de ce projet [11]. Comme outil d'apprentissage auto-régulé, il doit permettre à l'utilisateur-apprenant de décider de l'organisation de son travail et donc d'interrompre son activité quand il le souhaite. Ainsi *Pensum* associe à chaque texte-source (ou groupe de textes-sources), l'état courant de la synthèse de l'utilisateur, ainsi que l'ensemble des rétroactions proposées. Les rétroactions sont de trois types et s'appuient toutes sur une mesure de proximité sémantique entre phrases, calculée avec LSA. Le système compare donc chaque phrase de la synthèse avec la suivante pour évaluer la cohérence du texte (¶ ou || dans la Fig. 1). Il compare toutes les phrases des textes sources avec celles de la synthèse pour inférer les phrases qui n'ont pas été prises en compte par l'apprenant (phrases en rouge dans la partie cours de la Fig. 1). Enfin, chaque phrase de la synthèse est comparée avec les phrases des textes-sources pour détecter les hors-sujet (phrases en rouge dans la partie synthèse). À la demande de l'apprenant et selon un coefficient de sévérité qu'il contrôle, la synthèse est analysée et « décorée » avec les informations précédemment nommées.

Étant donnée la difficulté à traiter le sens d'un texte automatiquement et l'ambition de favoriser la réflexivité, le système permet à l'apprenant de remettre en question toute rétroaction. Les rétroactions de type « oubli » peuvent être refusées parce que l'apprenant estime que la phrase non prise en compte n'est pas pertinente pour la synthèse en cours, auquel cas elle est grisée, lui permettant de se focaliser sur le reste du contenu. Si toutefois, il est en désaccord avec le système, l'opportunité de justifier sa réponse lui sera fournie en lui demandant de lier explicitement la phrase incriminée avec celle de la synthèse qui lui correspond (phrase bleutée et surlignée dans la Fig. 1). Le même type de dispositif est proposé pour les phrases de la synthèse, à ceci près que si l'apprenant ne se justifie pas, la phrase ne sera pas grisée, mais bleutée pour indiquer qu'une rétroaction la concernait et qu'elle a été refusée. De plus, pour les sauts de cohérence, l'apprenant n'aura pas à justifier sa réponse. Enfin, pour permettre une activité tutorée et une intégration du flux d'information entre *widgets* de la plate-

forme, l'apprenant peut fournir un flux RSS à un enseignant, qui contiendra toutes les versions successives de la synthèse. Les rétroactions contestées peuvent ainsi être le support d'interactions.



**Fig. 1.** Interface de *Pensum* après traitement des rétroactions par l'apprenant. Le champ du haut contient une section du cours à lire. Le champ du bas contient le résumé ou la synthèse produite par l'apprenant suite à la lecture du cours. Les rétroactions de *Pensum* sont codées en couleurs.

#### 4 *ReaderBench*, un EPA pour analyser la compréhension en lecture et calibrer des textes suivant leur complexité

L'utilisation de stratégies pendant la lecture est un déterminant important de la compréhension [12] chez les adultes comme chez les enfants [9]. Ces stratégies désignent la capacité qu'ont les lecteurs d'exercer une veille attentive sur ce qu'ils comprennent au fur et à mesure de leur lecture, et de mettre œuvre des procédures de régulation en cas de difficulté [13]. Elles peuvent être recueillies par des auto-explications et ont été catégorisées par [10] en régulation de la compréhension (*monitoring*), paraphrase, élaboration (construction d'une nouvelle connaissance inférentielle à partir du texte), prédiction et *bridging* (relation entre deux parties du texte, ou entre une partie du texte et une connaissance antérieure). Notre premier but, avec *ReaderBench* [14], est

d'analyser automatiquement l'emploi de ces stratégies à partir des auto-explications des élèves (transcrites de l'oral) en utilisant des méthodes de TAL inspirées du travail de McNamara [10, 15].

La Fig. 2 est une copie d'écran de *ReaderBench* rendant compte des mesures de cohésion (agrégées à partir de distances sémantiques obtenues à partir d'ontologies [16], LSA [7] et LDA, pour *Latent Dirichlet Allocation* [17]) entre les paragraphes du texte à lire (dernière colonne de droite), et des stratégies de lecture pour chaque auto-explication (en gris), codées ainsi : **contrôle**, **causalité**, **paraphrase** [entre crochets, un numéro d'index référant aux mots du texte lu], **concept inféré** [\*] and **bridging** entre phrases du texte lu.

Le fonctionnement de LSA est maintenant bien connu et nous ne le détaillerons pas ici. LDA permet de déterminer, dans une approche « paquets de mots », la structure en « thèmes » (*topics*) d'un document textuel, partant du principe que tout texte est une combinaison d'un petit nombre de thèmes (valeur arbitraire de 100), dont on peut identifier la présence de manière probabiliste. Chaque mot du document est ainsi intégré dans les 100 thèmes principaux, assorti d'une probabilité d'appartenance à ce thème. Chaque nouveau document traité peut à son tour être analysé et sa probabilité d'appartenir au différents thèmes est évaluée. Les scores de cohésion ont une validité améliorée par l'agrégation de multiples mesures de similarité (LSA, LDA et Wu-Palmer), ce qui améliore la mesure de cohésion entre paragraphes adjacents [14]. Ainsi, cette combinaison de mesures (ontologies, espaces vectoriels et modèles probabilistes) permet de réduire les erreurs induites par une mesure unique.

L'autre fonction principale de *ReaderBench* décrite ici est l'évaluation de la complexité textuelle des textes analysés. Cette évaluation est une tâche très difficile compte tenu, d'une part, de la difficulté pour un enseignant de se mettre à la place du niveau de compréhension et de vocabulaire d'un élève et, d'autre part, de la difficulté de comparer un ensemble de textes selon ce paramètre. Néanmoins, pouvoir sélectionner du matériel de lecture qui soit accessible à un niveau scolaire donné est très important pour l'enseignant [18].

Une analyse de la complexité textuelle sur un grand nombre de dimensions est disponible (voir leur description complète dans [5, 14]), dont les résultats font l'objet d'une classification avec des machines à vecteur de support (*Support Vector Machines* [19]), très appropriées pour ce type de données [20, 21]. Les dimensions choisies couvrent un grand nombre de paramètres, des formules classiques de lisibilité, les techniques issues de l'évaluation automatique de copies, ainsi que des facteurs morpho-syntaxiques [5] et sémantiques [14]. La Fig. 3 est une copie d'écran des informations sur la complexité textuelle, le choix des paramètres étant libre. Comme aucun corpus n'était disponible pour la langue française, nous avons validé cette méthode avec le corpus TASA (*Touchstone Applied Science Associates*) en utilisant les scores DRP (*Degree of Reading Power*, dont la valeur correspond approximativement au niveau de classe) intégrés à chaque texte, et extrait d'environ 1 000 documents rangés en 6 classes de difficulté croissante [22]. La prise en compte de toutes les métriques de complexité de *ReaderBench* montre que les résultats des SVM prédisent correctement les scores DRP (*Accord Exact* = .763 and *Accord Adjacent* = .997, resp. pour une prédiction exacte et un score ayant un écart inférieur à 1 avec le score exact). Une

deuxième comparaison a été effectuée avec seulement les métriques de haut niveau (morphologie et sémantique), afin d'éviter une certaine circularité, le score DRP étant fondé sur des facteurs de bas niveau (*Accord Exact* = .597 and *Accord Adja-cent* = .943).

Text	Causality	Control	Paraphr...	Knowle...	Bridging	Cohesion
la mère[8] devint toute blanche . elle dit[5] à son mari il y a quelqu'un dans la maison[2] . ils arrêtèrent[9] tous de manger[10] . ils étaient tous sur le qui - vive , la voix[7] reprit[11] salut[6] , salut[6] , salut[6] . le frère[12] se mit à crier ça recommence[13] ! matilda se leva et alla éteindre la télévision[3] .						0.315
Je ai compris[4] que c'est une famille[2] la famille[2] dans laquelle il ? suis qui dînent[1] devant la télé[3] . et qui . tout de un coup il z entendent[4] une voix[7] qui leur dit[5] salut[6] . et du coup ils ont peur donc parce que la mère[8] de matilda ? donc c'est qu'ils pensent que ils ont peur . alors ils arrêtent[9] de manger[10] . puis le frère[12] commence à comprendre quelque chose en disant ça recommence[13]	5	1	13	0	1	
la mère . paniquée . dit à son mari : henri . des voleurs[15] . ils sont dans le salon . tu devrais[14] y aller . le père , raide sur sa chaise ne bougea pas . il n'avait pas envie de jouer au héros . sa femme lui dit : alors , tu te décides ? ils doivent[14] être en train de faucher l'argenterie[16] !						0.294
alors le père que c'est une famille[2] peut - être assez riche parce que il y a de l'argenterie[16] . et qui pensent que ceux qui doit[14] être riche ou que y'a beaucoup de voleurs[15] dans notre dans leur maison avec	2	1	3	1	1	
monsieur verdebois s'essuya nerveusement les lèvres avec sa serviette et proposa d'aller[17] voir[18] tous ensemble . la mère attrapa un tisonnier au coin de la cheminée . le père[19] s'arma d'une canne de golf posée dans un coin . le frère attrapa un tabouret . matilda prit[9] le couteau avec lequel elle mangeait . puis ils se dirigèrent tous les quatre vers la porte du salon en marchant sur la pointe des pieds .						0.399
à ce moment - là , ils entendirent à nouveau la voix . matilda fit alors irruption dans la pièce en brandissant son couteau et cria haut[20] les mains[21] , vous êtes pris[9] ! les autres la suivirent en agitant leurs armes .						0.189
donc la c'est un peu déjà comment s'appelle la famille . et puis ils racontèrent que là vu que le père[19] veut pas y aller[17] tout seul . il est accompagné de toute sa famille pour aller[17] voir s'y a un voleur . et y a la la parole[1] ça le bruit aussi ? qui recommence . et du coup elle . la petite fille[1] qui s'appelle matilda commence à avoir peur . donc elle lui dit haut[20] les mains[21] vous êtes pris[9]	4	2	5	2	1	

Fig. 2. Analyse des stratégies de lecture et de la cohésion dans *ReaderBench*.

Ces différentes fonctionnalités peuvent stimuler l'apprentissage auto-régulé. Tout d'abord *ReaderBench* procure aux apprenants une meilleure idée des caractéristiques (mots-clés, complexité) d'une série de documents qu'ils projettent de lire (*niveau objet*, voir § 2). Ensuite, ils peuvent utiliser l'analyse des auto-explications pour évaluer leurs notes ou leur avis à propos des textes (*niveau méta*). Ces informations, notamment sur les textes, sont également utiles aux enseignants, pour les aider à présenter des documents de caractéristiques et difficulté en adéquation avec le niveau de leurs élèves.

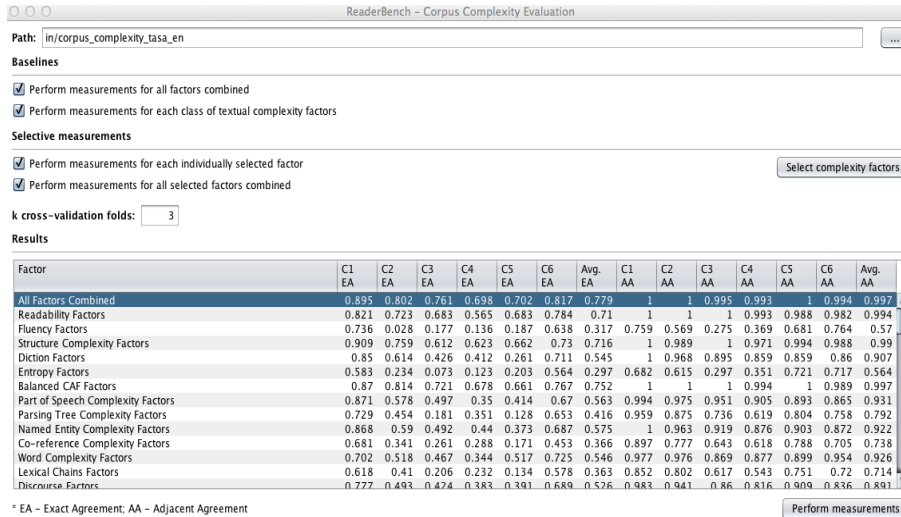


Fig. 3. Analyse de la complexité textuelle dans *ReaderBench*.

## 5 Architectures et Méthodes

L'architecture de *Pensum*, représentée dans la Fig. 4 suivante, est classique et les méthodes TAL utilisées dans *Pensum* sont relativement peu nombreuses et simples : un script *Perl* de segmentation en phrases, un script supprimant les mots outils (non représenté sur la figure) et le recours à *R-LSA*. Une base de données (au centre) stocke toutes les données nécessaires : les cours à synthétiser, les synthèses produites, les informations sur l'utilisateur, les notes, les rétroactions et les actions de l'utilisateur sur ces dernières [23]. Un module d'affichage récupère de la base de données un document de cours à afficher à l'apprenant, ainsi que les possibles précédentes rétroactions. L'apprenant a deux principaux moyens d'agir sur *Pensum*. Tout d'abord, *éditer la synthèse*, dont les phrases sont détectées automatiquement et ensuite envoyées à un service web utilisant *R-LSA* (une version de LSA implémentée avec le logiciel de traitements statistiques) [24] afin de réaliser les différents calculs de similarité sémantique. Il peut aussi *agir sur les rétroactions* en les contestant (p. ex., « Je pense, contrairement à ce qu'affiche *Pensum*, que ces deux phrases sont cohérentes »), et ces dernières sont à leur tour envoyées au module d'affichage.

L'architecture de *ReaderBench* est plus complexe et de nombreux traitements sur les documents fournis en input s'enchaînent, organisés comme décrit en Fig. 5 ci-dessous, mais qu'il serait trop long de présenter en détail ici [25].



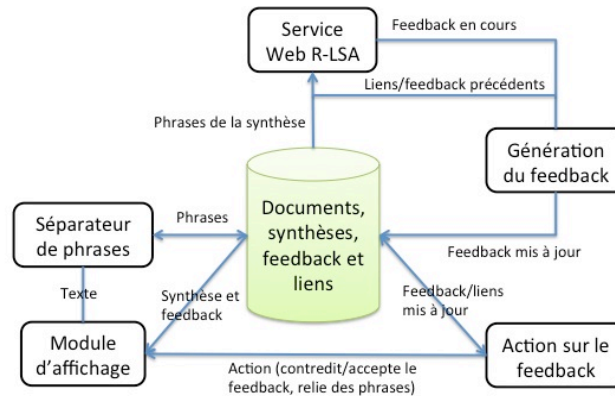


Fig. 4. Architecture de *Pensum* [9, p. 12].

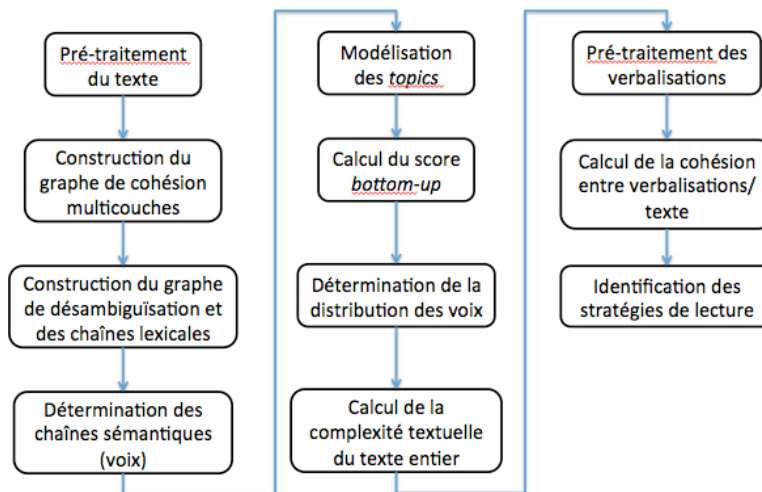


Fig. 5. Flux des traitements automatiques de la langue de *ReaderBench*.

## 6 Quelques réflexions sur les méthodes

Les techniques de TAL mises en œuvre dans ces deux EPA sont d'un niveau de sophistication très différent : celles mises en œuvre dans *Pensum* sont simples (comparaisons sémantiques avec LSA, sans autre traitement complexe), celles mises en œuvre dans *ReaderBench* sont bien plus nombreuses et complexes, et tirent bénéfice de techniques de calcul parallèle [26] et d'une puissance accrue du matériel informatique. Indépendamment de la technique et de la puissance de traitement, nous pouvons évoquer quelques questions et problèmes.

Tout d'abord, il faut questionner l'adéquation théorique des modèles sous-jacents à ces techniques de TAL et son incidence sur les sorties des systèmes les utilisant. Ces techniques autorisent une analyse des documents par rapport à leur modèle linguistique sous-jacent, qui ne peut être qu'une approximation, un point de vue sur la langue. L'une des difficultés est d'évaluer l'adéquation de ce point de vue avec la tâche à traiter.

Par exemple, l'une des limites de LSA, tel qu'il est utilisé dans *Pensum*, est également l'un de ses avantages. Le point de vue des auteurs de LSA sur la langue, si l'on s'écarte de sa représentation mathématique, pourrait être décrit comme évaluant, dans une approche paquets-de-mots, la *thématique* d'un mot ou ensemble de mots, exprimée en coordonnées d'un vecteur dans un espace vectoriel multidimensionnel, préalablement créé par l'analyse d'un grand corpus de textes exprimant des connaissances générales (e.g., encyclopédie, journaux). La proximité sémantique de deux vecteurs, exprimée par le cosinus de l'angle qu'ils forment, est donc influencée par la position des mots les uns par rapport aux autres dans le corpus général. Le fait de modéliser ainsi la langue permet de développer des ressources *ad hoc* à relativement moindre coût — pour une langue de spécialité donnée constituer un corpus est une tâche réalisable en quelques mois, par rapport à la description du système qui régit son contenu. Cela favorise ainsi la conception de systèmes multilingues—il a été très aisé de réaliser une version de *Pensum* traitant de documents en anglais.

Mais ce point induit trois problèmes, limitant les possibilités de modélisation sémantique. Tout d'abord et par exemple, par leur fréquence dans la langue, les adverbes de négation ne peuvent être pris en compte, ainsi la polarité ne peut être prise en compte dans les évaluations. De la même manière, une lexie et son pluriel seront des entités différentes et leur proximité sémantique ne sera que la conséquence de l'entraînement du système. On peut donc se demander si, dans le cas de *Pensum*, qui décrit finalement le sens comme une thématique, les résultats ne bénéficieraient pas d'un outil de lemmatisation (identification de la forme canonique des mots : « *apprends* » et « *apprenaient* » également remplacés par « *apprendre* »). C'est d'ailleurs le parti-pris de la conception de *ReaderBench*, qui a une approche qu'on pourrait nommer « LSA + étiquettes » (*Tagged LSA*) incluant, dans la première étape de la Fig. 5, une chaîne de traitements préliminaires au traitement par LSA : suppression de mots-outils, lemmatisation, étiquetage grammatical (*POS tagging*), permettant une meilleure prise en compte du sens des phrases traitées, au détriment d'une vitesse d'analyse moindre.

Un deuxième problème est lié à la comparaison sémantique : elle ne donne des résultats valides qu'avec un nombre de mots suffisamment important (voir [27] pour une discussion), mais comparer de trop grands ensembles de mots donne une idée trop grossière de leurs caractéristiques (notamment celles liées à la cohésion, qui est au cœur d'un grand nombre de fonctionnalités de *ReaderBench*, voir [28]). Les auteurs de cette dernière référence préconisent une analyse plus précise de la structure du texte, comme celle mise en œuvre dans *ReaderBench*.

Un troisième problème est lié à la dépendance directe des données du corpus de connaissances générales : la difficulté de traiter la grande variété des productions d'étudiants influence l'évolutivité du système. Il ne suffit pas d'améliorer la représen-

tativité du corpus de connaissances générales pour améliorer les résultats du système. En effet, la modification de ce corpus entraîne la modification de toutes les mesures effectuées par le système et demande une « recalibration » des valeurs-seuils. Les mesures automatiques de nos systèmes utilisent en effet de nombreux seuils, dont le niveau contient une assez grande partie d'arbitraire. Si des validations avec du matériel évalué par d'autres méthodes (*e.g.*, humaines) permettent de les ajuster, la grande variabilité des textes donnés à lire et des productions des apprenants fait qu'il est impossible d'avoir des ajustements automatiques universels. La solution du curseur de sévérité de *Pensum* règle en partie ce problème, et *ReaderBench*, jusqu'ici testé avec des textes narratifs de littérature de jeunesse, devra également être testé sur d'autres types de textes (*e.g.*, des textes argumentatifs).

Ces problèmes sont difficiles à régler et nous ne prétendons pas les avoir résolus. Deux pistes futures s'annoncent : d'une part, la constitution de banques de données (textes-sources et exemples de productions pré-étiquetées). D'autre part, et pour évaluer la complexité textuelle, la constitution de corpus importants en taille et dont les documents ont été classés par des humains plutôt que par des scores de lisibilité automatique. Ce type de corpus est difficilement trouvable ou constituable, ce qui rend les mesures de complexité en français par *ReaderBench* non encore validées.

## 7 Conclusion

Les méthodes complexes utilisées dans *Pensum* et *ReaderBench* rendent leur conception et leur réglage difficiles. De plus, leur utilisation est peu évidente sans information précise sur la manière dont ils fonctionnent, voire sans formation [4]. En effet, les utilisateurs (enseignants ou apprenants), peuvent difficilement faire des analogies avec des logiciels existants et d'usage courant, ce qui rend le comportement des logiciels peu prédictible. De plus, leurs très nombreuses fonctionnalités sont accessibles librement, il faut scénariser leur utilisation dans des sortes de flux de travail auto-régulés pour s'assurer de leur bien-fondé. Ce type de validation sera mené prochainement avec *ReaderBench*, ainsi que des validations l'insérant dans divers scénarios pédagogiques.

## Note des auteurs

La section 4 de cet article est une adaptation en français d'une partie de [14]. Ces recherches ont bénéficié des financements de la Communauté européenne (7<sup>e</sup> PCRD, projet LTfLL) pour *Pensum* et du projet 264207 ERIC (*Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1* pour *ReaderBench*. *ReaderBench* a également été en partie financé par l'ANR (projet DEVCOMP, ANR-10-BLAN-1907-01), et le premier auteur a bénéficié d'une bourse doctorale POSDRU/107/1.5/S/76909 (ValueDoc).

## 8 Références

1. Fiedler, S.H.D. and Våljataga, T.: Personal learning environments: concept or technology? *International Journal of Virtual and Personal Learning Environments*, 2(4), 1–11 (2011)
2. Henri, F. and Charlier, B.: Personal Learning Environment: a concept, an application, or a self-designed instrument? In: *Int. Conf. on Information Technology Based Higher Education and Training (ITHET 2010)*, pp. 44–51, Cappadocia, Turkey (2010)
3. Dabbagh, N. and Kitsantas, A.: Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. *Internet and Higher Education*, 15(1), 3–8 (2012)
4. Loiseau, M., Dupré, D. and Dessus, P.: *Pensum*, un système d'aide à la compréhension de cours à distance. In: Bétrancourt, M., Depover, C., Luengo, V., Lièvre, B.D. and Temperman, G. (eds.) *Conférence EIAH 2011*, pp. 287–299. Presses de l'université de Mons, Mons (2011)
5. Dascalu, M., Trausan-Matu, S. and Dessus, P.: Towards an integrated approach for evaluating textual complexity for learning purposes. In: Popescu, E., Klamma, R., Leung, H. and Specht, M. (eds.) *Advances in web-based learning (ICWL 2012)*, Vol. LNCS 7558, pp. 268–278. Springer, New York (2012)
6. Dascalu, M., Trausan-Matu, S. and Dessus, P.: Cohesion-based analysis of CSCL conversations: Holistic and individual perspectives. *10th Int. Conf. Computer-Supported Collaborative Learning (CSCL 2013)*. Univ. of Wisconsin, Madison (2013)
7. Landauer, T.K. and Dumais, S.T.: A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychol. Rev.*, 104(2), 211–240 (1997)
8. Vovides, Y., Sanchez-Alonso, S., Mitropoulou, V. and Nickmans, G.: The use of e-learning course management systems to support learning strategies and to improve self-regulated learning. *Educational Research Review*, 2(1), 64–74 (2007)
9. Nardy, A., Bianco, M., Toffa, F., Rémond, M. and Dessus, P.: Contrôle et régulation de la compréhension : L'acquisition de stratégies de 8 à 11 ans. In: David, J. and Royer, C. (eds.) *L'apprentissage de la lecture : convergences, innovations, perspectives*. Peter Lang, Berne (sous presse)
10. McNamara, D.S.: SERT: Self-Explanation Reading Training. *Discourse Proc.*, 38, 1–30 (2004)
11. Dessus, P., Lemaire, B., Loiseau, M., Mandin, S., Villiot-Leclercq, E. and Zampa, V.: Automated free-text assessment: Some lessons learned. *International Journal of Continuing Engineering Education and Life-Long Learning*, 21(2/3), 140–154 (2011)
12. Graesser, A.C.: An introduction to strategic reading comprehension. In: McNamara, D.S. (ed.) *Reading comprehension strategies: theories, intervention and technologies*, pp. 3–26. Erlbaum, Mahwah (2007)
13. Bianco, M.: La compréhension de textes : peut-on l'apprendre et l'enseigner ? In: Crahay, M. and Dutrévis, M. (eds.) *Psychologie des apprentissages scolaires*, pp. 230–256. De Boeck, Bruxelles (2010)
14. Dascalu, M., Dessus, P., Trausan-Matu, S., Bianco, M. and Nardy, A.: ReaderBench, an environment for analyzing text complexity and reading strategies. *16th Int. Conf. on Artificial Intelligence in Education (AIED 2013)*, Memphis (2013)

15. McNamara, D., Boonthum, C. and Levinstein, I.: Evaluating self-explanations in iSTART: Comparing word-based and LSA algorithms. In: Landauer, T.K., McNamara, D., Dennis, S. and Kintsch, W. (eds.) *Handbook of Latent Semantic Analysis*, pp. 227–241. Erlbaum, Mahwah (2007)
16. Sagot, B.: Wordnet Libre du Francais (WOLF). INRIA, Paris (2008)
17. Blei, D.M., Ng, A.Y. and Jordan, M.I.: Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5), 993–1022 (2003)
18. Nelson, J., Perfetti, C., Liben, D. and Liben, M.: Measures of text difficulty. Technical Report to the Gates Foundation (2011)
19. Cortes, C. and Vapnik, V.N.: Support-Vector Networks. *Machine Learning*, 20(3), 273–297 (1995)
20. François, T. and Miltsakaki, E.: Do NLP and machine learning improve traditional readability formulas? In: *First Workshop on Predicting and improving text readability for target reader populations (PITR2012)*, pp. 49-57. ACL, Montreal (2012)
21. Petersen, S.E. and Ostendorf, M.: A machine learning approach to reading level assessment. *Computer Speech and Language*, 23, 89–106 (2009)
22. McNamara, D.S., Graesser, A.C. and Louwerse, M.M.: Sources of text difficulty: Across the ages and genres. In: Sabatini, J.P. and Albro, E. (eds.) *Assessing reading in the 21st century*. R&L Education, Lanham (in press)
23. Trausan-Matu, S., Dessus, P., Rebedea, T., Loiseau, M., Dascalu, M., Mihaila, D., Braidman, I., Armitt, G., Smithies, A., Regan, M., Lemaire, B., Stahl, J., Villiot-Leclercq, E., Zampa, V., Chiru, C., Pasov, I. and Dulceanu, A.: Deliverable 5.3 LTfLL – Learning support and feedback. OUNL, Heerlen (2011)
24. Wild, F.: An LSA package for R. In: *Mini-Proceedings of the 1st European Workshop on Latent Semantic Analysis in Technology-Enhanced Learning*, pp. 12 (2007)
25. Dascalu, M.: Analysing discourse and text complexity for learning and collaborating. Univ. Grenoble Alpes/Univ. Polytechnica Bucharest, Grenoble/Bucharest, Unpublished PhD Thesis (2013)
26. Dascalu, M., Dobre, C., Trausan-Matu, S. and Cristea, V.: Beyond traditional NLP: A distributed solution for optimizing chat processing. *10th International Symposium on Parallel and Distributed Computing (ISPDC 2011)*, Cluj-Napoca, Romania (2011)
27. Villalon, J. and Calvo, R.A.: Single document semantic spaces. In: *8th Australasian Data Mining Conference (AusDM'09)*, Melbourne (2009)
28. Bestgen, Y.: Evaluation automatique de textes et cohésion lexicale. *Discours*, 11, (2012)