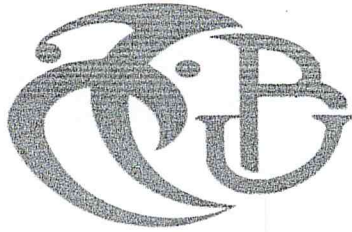
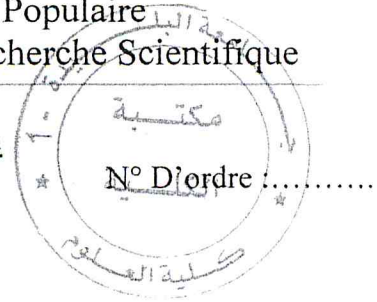


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Bouzegzeg Djamel-Eddine

Boudehane Sid ali

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

Sujet :

Module d'évaluation automatique de réponse courte dans MOODLE

Soutenu le :

M. Sidoumou, M
M. Chikhi, I
M. Cherfa, N
M. Bennouar Djamel

Président
Examineur
Examineur
Promoteur

Promotion
2014 / 2015

Remerciements

Nous remercions d'abord Dieu Le Tout Puissant et Le Tout Miséricordieux qui nous a donné la force, la patience, et l'abstinence de continuer jusqu'au bout.

La première personne que nous tiendrons à remercier est notre Promoteur Dr. BÉNOUÛR Djamal, pour l'orientation, la confiance et la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de s'enrichir par leurs propositions. Enfin, je tiens également à remercier l'ensemble de la famille enseignante de l'université SAAD DAHLÛB De Blida et toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

A mes parents qui se sont sacrifiés pour moi pour que je finisse mon projet, mes études, pour que je sois leur fierté, à ma chère mère qui était là pour moi, avec ses conseils, sa gentillesse, et son temps, que Dieu la protège. Ainsi à mon cher père qui m'a libéré de toutes les responsabilités pour que je termine mes études librement, que dieu les protège.

A ma sœur et son mari qui ont été là pour moi depuis tout le début, qui m'ont poussé avec leurs encouragements un peu durs parfois, c'est bien d'avoir une telle famille.

A tous mes amis qui m'ont aidé soit par leur encouragements, ou par quoi ce soit, des amis qui veillent sur toi, qui t'oublient pas et qui veulent que aie tout le bonheur du monde.

A tous les enseignants qui m'ont appris beaucoup de choses depuis le primaire jusqu'à la dernière année de mon cursus scolaire en Master 2.

Djamel-Eddine Bouzegzeg



Dédicaces

A mon père qui s'est sacrifié afin que rien ne m'empêche de réaliser le meilleur. Ce travail est le fruit des sacrifices que tu as consenti pour mon éducation. Puisse Dieu, le tout Puissant, te préserver et t'accorder santé, longue vie et bonheur.

A ma très chère mère: Tu es l'exemple de dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A toute ma famille, à tous mes amis: Madjid, Zakaria, Youcef, Nassim, Raouf : je vous remercie de votre patience, vous qui m'avez toujours aidé à avancer, vous êtes tous des amis fidèles et si gentils, merci d'être restés toujours à mes côtés dans toutes les circonstances.

A mes enseignants de l'école primaire jusqu'à l'université : merci pour vos conseils précieux Qui m'ont guidé à réaliser un parcours scolaire que je considère réussi. Mes chers enseignants et seconds parents permettez-moi de vous exprimer l'expression de mes sentiments les plus respectueux.

Boudechane Sidali

Tables

Des matières

Introduction	1
2. Problématique :	2
3. Objectifs :	3
4. Organisation du document :	3
I. Etat de l'art	5
I.1. Introduction :	6
I.2. Similarité syntaxique ou Similarité entre chaîne de caractères :	7
I.2.1. La similarité exacte :	7
I.2.2. La distance de similarité :	9
I.3. Similarité sémantique :	10
I.4. Evaluation des approches et systèmes actuels :	Erreur ! Signet non défini.1
II. Solution proposé	14
II.1. Introduction générale au système proposé :	15
II.2. Architecture général du système :	15
II.3. Les contrôleurs et directives de la première phase:	18
II.3.1. Le contrôleur de choix des seuils et des algorithmes de similarité textuelle :	19
II.3.2. Contrôleur de gestion des séparateurs :	19
II.3.3. Contrôleur de gestion des caractères.	20
II.3.3.1. Option : Les accents :	20
II.3.3.2 Option : la casse :	21
II.3.3.3. Option dédoublement de caractère :	21
II.3.4. Contrôleur de gestion des mots clé :	22
II.3.5. Contrôleur de gestion des déterminants :	22
II.3.6. Contrôleur de gestion des synonymes :	23
II.3.7. Contrôleur de Gestion des homophones :	23
II.3.8. Les fautes d'orthographe :	23
II.3.9. Contrôleur de gestion du contenu dynamique :	24
II.3.10. Les abréviations :	24
II.4. La deuxième composante du système :	24
II.5. Conclusion :	25



III. Implémentation et tests :	26
III.1. Introduction	27
III.2. L'environnement de développement:.....	27
III.2.1. WAMP :.....	27
III.2.2 Le serveur web :.....	28
III.2.3. Apache HTTP Server:.....	29
III.2.4.Langages web :	30
III.2.4.1. HTML :	30
III.2.4.2. CSS	30
III.2.4.3. PHP :	31
III.2.4.JavaScript :.....	32
III.2.5.Langage d'interrogation de base de données :.....	33
III.2.5.1. MYSQL :	33
III.2.6. La base de données utilisée :	33
III.2.6.1. Schéma relationnel de la BD :	33
III.3. Les algorithmes relatifs aux directives et indications de l'enseignant :	35
III.3.1. Ignorer espaces :	35
III.3.2. Désactiver la sensibilité à la casse :.....	36
III.3.3. Ignorer les déterminants :	37
III.3.4.Ignorer les accents :	37
III.3.5.Eliminer les doublons :	37
III.3.6. Utiliser Les synonymes :	38
III.3.7. Reconnaître Les Abréviations :	40
III.3.8. Gestion des mots clés :	41
III.3.9. Correction des fautes d'orthographe :	42
III.3.10.Supprimer/Remplacer caractère :	42
III.3.11. Les références (Corpus) :.....	43
III.3.12. Correction d'une référence :	44
III.3.12. Correction d'une réponse :.....	47
III.4. L'application autonome de correction de réponses courtes:	48
III.4.1.Inscription :	48
III.4.2.Connexion :	50
III.4.3.L'enseignant :	50
III.4.3.1.Création d'examen :	51

III.4.3.2. Edition des questions :	51
III.4.3.2.1. Le contenu dynamique :	53
III.4.3.2.2. Synonymes :	54
III.4.3.2.3. Absence des mots :	54
III.4.3.2.4. Mots Clé :	55
III.4.3.2.5. Négliger les blancs :	55
III.4.3.2.6. Abréviation :	56
III.4.3.2.7. Remplacer/supprimer des caractères :	56
III.4.3.2.8. Choix de la référence :	57
III.4.3.2.9 Remarque :	58
III.4.3.3. Correction d'examen :	58
III.4.3.4. Edition des examens/questions/références :	59
III.4.3.4.1. Supprimer une référence :	60
III.4.3.4.2. Editer une référence :	61
III.4.3.4.3. Ajouter une référence:	61
III.4.3.4.4 Edition d'une question :	62
III.4.4. Administration :	64
III.4.5. Passer un examen :	65
III.4.6. Se déconnecter :	65
III.5. Le module MOODLE :	66
III.5.1. Introduction à l'API de création de formulaire (formslib) :	66
III.5.1.1. Header :	67
III.5.1.2. Textfield :	67
III.5.1.3. Checkbox :	67
III.5.1.4. Liste déroulante :	68
III.5.1.5. Bouton :	68
III.5.1.6. La méthode get_string() :	68
III.5.1.7. Autres méthodes :	68
III.5.2. La base de données MOODLE :	69
III.5.3. Remarques :	69
III.5.4. Correction d'une question :	70
III.5.5. Interfaces du module de MOODLE :	70

III.5.6. Difficultés rencontrés Moodle :	76
III.6. Conclusion :	77
IV. Conclusion :	78
IV.1. Conclusion :	79
IV.2. Perspectives :	Erreur ! Signet non défini.

Bibliographie

Annexes

Tables des figures

Figure 3. 1 : Architecture générale du système	16
Figure 3. 2 : Création du corpus	17
Figure 3. 3 : schéma de la deuxième partie	18
Figure 4. 1 : code d'éliminer espace (1/3)	35
Figure 4. 2 : code d'éliminer espace (3/3)	36
Figure 4. 3 : code d'éliminer espace (3/3)	36
Figure 4. 4 : Code de «éliminer la casse ».	36
Figure 4. 5 : Code de « éliminer déterminants ».	37
Figure 4. 6 : code de « éliminer les accents ».	37
Figure 4. 7 : code « éliminer les doublons »	37
Figure 4. 8 : fichier XML contenant les synonymes et leurs pénalités.	38
Figure 4. 9 : code de génération du dictionnaire XML de synonyme.	39
Figure 4. 10 : code d'ajout des synonymes à la liste des références.	39
Figure 4. 11 : code récupération des abréviations de la BD.	40
Figure 4. 12 : code d'ajout des abréviations à la liste des références.	40
Figure 4. 13 : code récupération des mots à supprimer de la BD.	41
Figure 4. 14 : code d'ajout des mots à supprimer à la liste des références.	41
Figure 4. 15 : code de calcul de nombre de faute pour un cas.	42
Figure 4. 16 : code de calcul de nombre des fautes pour tous les cas.	42
Figure 4. 17 : code de récupération des caractères à supprimer/remplacer.	42
Figure 4. 18 : code des caractères à supprimer/remplacer.	43
Figure 4. 19 : code de l'écriture des références.	43
Figure 4. 20 : fichier XML contenant les références et leurs pénalités.	44
Figure 4. 21 : Code de « correction de référence »	44
Figure 4. 22 : Code de « correct » (1/3).	45
Figure 4. 23 : Code de « correct » (2/3)	46
Figure 4. 24 : Code de « correct » (3/3)	46
Figure 4. 25 : Code de « correction de question »	47
Figure 4. 26 : page d'accueil de l'application	48
Figure 4. 27 : interface d'inscription (1/3)	48
Figure 4. 28 : interface d'inscription (3/3)	49
Figure 4. 29 : interface de connexion	50
Figure 4. 30 : interface de l'enseignant « Menu d'outils »	50
Figure 4. 31 : Interface de création d'un examen	51
Figure 4. 32 : Interface de création d'une question (1/2).	52
Figure 4. 33 : Interface de création d'une question (2/2).	52
Figure 4. 34 : interface des mots dynamiques.	53
Figure 4. 35 : interface des synonymes.	54
Figure 4. 36 : interface d'absence des mots	54
Figure 4. 37 : interface des mots clé	55
Figure 4. 38: interface de « négliger les blanc ».	55
Figure 4. 39 : interface des abréviations.	56

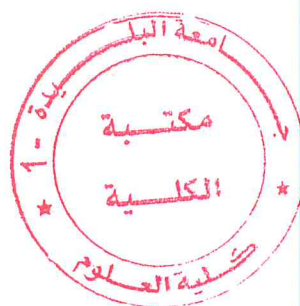


Figure 4. 40: interface de « remplacer/supprimer caractère».	56
Figure 4. 41 : choix des options de la référence (1/2).	57
Figure 4. 42 : choix des options de la référence (2/2).	57
Figure 4. 43 : Interface de correction des examens.	58
Figure 4. 44 : interface de choix de l'examen.	58
Figure 4. 45 : interface de choix de question.	59
Figure 4. 46 : choix de la référence à supprimer.	59
Figure 4. 47 : choix de la référence à modifier.	60
Figure 4. 48 : interface d'édition d'une question.	61
Figure 4. 49 : interface de suppression des questions.	61
Figure 4. 50 : interface d'ajout d'une question.	62
Figure 4. 51 : interface de suppression des examens.	62
Figure 4. 52 : interface de modification d'un examen.	63
Figure 4. 53 : interface de validation des étudiants (enseignants).	63
Figure 4. 54 : interface de l'examen pour l'étudiant	64
Figure 4. 55 : « edit_testd_form.php »	65
Figure 4. 56 : méthode de définition du formulaire	66
Figure 4. 57 : ajouter un header	66
Figure 4. 58 : ajouter un textfield	66
Figure 4. 59 : ajouter une liste déroulante	67
Figure 4. 60 : ajouter un bouton	67
Figure 4. 61 : le fichier « qtype_testd.php »	67
Figure 4. 62 : fermer un header	68
Figure 4. 63 : Exemple d'une table dans le fichier install.xml.	68
Figure 4. 64 : choix du type d'une question sous MOODLE.	70
Figure 4. 65 : Interface d'ajout d'une question sous MOODLE (1/13).	71
Figure 4. 66 : Interface d'ajout d'une question sous MOODLE (2/13).	71
Figure 4. 67 : Interface d'ajout d'une question sous MOODLE (3/13)	72
Figure 4. 68 : Interface d'ajout d'une question sous MOODLE (4/13)	72
Figure 4. 69 : Interface d'ajout d'une question sous MOODLE (5/13) (Négliger).	72
Figure 4. 70 : Interface d'ajout d'une question sous MOODLE (6/13)	73
Figure 4. 71 : Interface d'ajout d'une question sous MOODLE (7/13)	73
Figure 4. 72 : Interface d'ajout d'une question sous MOODLE (8/13) (Abréviations).	73
Figure 4. 73 : Interface d'ajout d'une question sous MOODLE (9/13)	74
Figure 4. 74 : Interface d'ajout d'une question sous MOODLE (10/13) (Fautes d'orthographe).	74
Figure 4. 75 : Interface d'ajout d'une question sous MOODLE (11/13) (Enregistrer la référence).	75
Figure 4. 76: Interface d'ajout d'une question sous MOODLE (12/13) (Ajouter une réponse).	75
Figure 4. 77 : Interface d'ajout d'une question sous MOODLE (13/13) (Enregistrer la question).	75

Figure 4. 78 : Interface de la liste des questions.

75

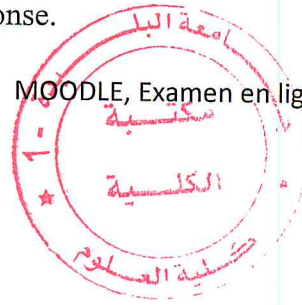
Figure 4. 79 : Interface de la correction d'une question.

76

RÉSUMÉ :

L'évaluation automatique des réponses écrites est un domaine de recherche très actif dans lequel le nombre de défis à relever est très important. L'évaluation automatique des réponses écrites tend à reproduire ce que fait un correcteur humain face à une réponse à une question. Ce travail propose un outil qui évalue les questions à réponses courtes. Contrairement aux outils existant, l'outil proposé donne un rôle très important à l'enseignant, qui est en réalité la seule personne à décider ce qui est important ou non dans une réponse. L'enseignant intervient dans le processus de préparation du système à l'évaluation automatique. Cette préparation se solde par la génération automatique d'un corpus pour chaque réponse.

Mots-clés : Réponse courte, Évaluation automatique, elearning, MOODLE, Examen en ligne.



ABSTRACT

Our work is a research theme that fits in the field of automatic evaluation of responses which is a subdomain of automatic processing of languages which in turn is a subdomain of the IA, the automatic evaluation of question aims to simulate as possible the human brain in the marking of questions or have a close result.

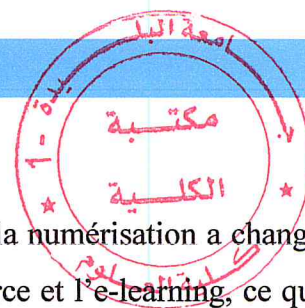
We offer a tool that evaluates the short-answer questions, our tool gives more control to the teacher, so the automatic correction will be more effective. Unlike other tool where the teacher only gives the answer without specifying rules to follow, the number of accepted answers is really reduced, where the slightest change could be fatal.

Our tool deals with different problems and situations that may occur, omission of accents, bad choice of a determinant, absence of a word which changes not the sense of the word, use a synonym Word, commit a simple misspelling or a mistake typing, etc. These cases often occur and if they are not treated so the number of accepted answers will be reduced.

The integration of our tool in high-end hostile platforms such as the platform MOODLE is a main goal, which will identify the result of our research that this tool may or may not be incorporated elsewhere.

KEYWORDS: Question, Automatic Evaluation, Assessment, Correction, MOODLE, Short Answer, Exam.

Introduction



Aujourd'hui, avec l'arrivée d'internet et l'explosion technologique, la numérisation a changé notre vue sur les différents processus de la vie, l'e-mail, l'e-commerce et l'e-learning, ce qui était de l'imagination est devenu réel, la technologie touche presque chaque aspect de notre vie.

Et le processus de l'enseignement et l'apprentissage n'échappe pas à cette loi, l'e-learning¹¹ (dont la définition sera évoquée plus tard) est disponible dans toutes les universités et les écoles du monde entier.

Quand on parle de l'enseignement et de l'apprentissage, on ne peut se passer de l'évaluation, une étape non négligeable de ce processus d'apprentissage.

Il existe deux types de questions d'évaluation largement répandus :

Les questions de type sélection : où l'étudiant est amené à choisir une réponse correcte parmi d'autres réponses, comme le choix multiple, VRAI/FAUX, arrangement des mots, remplir le vide, etc... (Utilisé principalement dans les branches médicales)

Les questions de type essai : où l'étudiant pourra utiliser ses propres mots pour répondre à la question, on en distingue deux catégories, question à réponses courtes (la catégorie qu'on va évoquer) où l'étudiant peut répondre avec une phrase sur une question qui aborde un aspect particulier, (utilisé dans les branches scientifiques et technologiques, dont l'informatique) et question essai où l'étudiant possède la liberté d'écrire un paragraphe et exprimer son opinion (utilisé dans les branches sociales, droits ,etc..).

Dans cette recherche on se focalisera sur les questions à réponses courtes, qui permettent – et d'une manière efficace – d'évaluer l'acquisition des connaissances, et permettre ainsi à l'enseignant de savoir les points acquis par l'étudiant et les difficultés qu'il pourrait avoir.

2. Problématique :

Quand il s'agit d'un modèle d'évaluation de type sélection, l'implémentation sera très facile, un code basique suffira largement.

¹¹ **Formation en ligne**, terme recommandé en France par la DGLFLF, (Journal officiel du 14/05/2005), voir France Terme.

Quant au type réponse courte (on ne parlera plus du type essai, ce n'est pas le cadre de notre recherche), La liberté donnée à l'étudiant pourrait engendrer des cas intraitables, pas avec une simple comparaison, on peut citer : l'utilisation d'un mot équivalent (synonyme), fautes d'orthographe (homophones), omettre les accents et d'autres cas qui seront détaillés plus tard.

Les outils de correction disponibles actuellement ne sont pas assez évolués pour traiter ce problème, en outre ils exigent des connaissances en informatique et surtout en expression régulières , on peut trouver des enseignants capables de manipuler ce genre d'outil , mais quand on arrive à des branches où l'informatique s'absente les choses se compliquent , l'outil doit être indépendant des capacités informatiques de l'enseignant , un outil qui peut être manipulé par quiconque qui peut toucher une souris. De plus ces outils ne peuvent guère deviner les fautes qui seront commises par l'étudiant : doublement de caractère, utilisation d'un mauvais déterminant, l'omission d'un accent, l'oubli d'un mot et le fait que ce mot soit indispensable ou non, etc.

3. Objectifs :

- Proposer une solution pour la correction automatique de réponses courtes.
- Mettre à la disposition de l'enseignant toutes les fonctionnalités nécessaires pour guider la correction automatique.
- Ajouter ce module à la plateforme MOODLE¹, et par conséquent maîtriser le code MOODLE et vérifier c'est ce dernier sera capable de supporter l'ajout d'un module complexe.

4. Organisation du document :

Chapitre I : Etat de l'art, où on va évoquer les solutions déjà présentes de la correction automatique, les critiquer si nécessaire et montrer leurs avantages et inconvénients, ainsi on peut en tirer ce qui est utile et éviter les erreurs préalablement commises, pour notre solution proposée.

Chapitre II : Solution proposée, notre solution proposée sera bien détaillée en cette partie, les fonctionnalités, la démarche utilisé étape par étape théoriquement.

Chapitre III : Implémentation et tests, notre solution proposée (théoriquement) sera implémentée et testée, étape par étape et sur deux plans : un module autonome de la solution proposée, et un module intégré sans la plateforme MOODLE, tout problème rencontré sera mentionné.

Chapitre IV : Conclusion, on conclura notre recherche par la vérification de l'utilité de notre solution proposée, et ainsi la vérification de la capacité de MOODLE à supporter des tels modules, on parlera ainsi de l'avenir de la correction automatique.

Bibliographie

Annexes

Etat de l'art

I.1. Introduction :

Depuis que l'e-learning est devenu indispensable à l'enseignement et l'apprentissage, les chercheurs ont essayé de concevoir un correcteur automatique de réponse qui est doté à la fois de l'intelligence de l'être humain et la rapidité de la machine. Cependant, le traitement automatique des langues n'a pas évolué à un point où on pourra se passer de l'être humain.

Dans le domaine de la correction automatique des réponses courtes, toutes les approches et outils se basent sur le calcul de la similarité entre une réponse de référence et la réponse de l'étudiant. La réponse de référence peut être une seule réponse ou un certain nombre de réponses organisées en corpus dans lequel nous trouvons des réponses types correctes et partiellement correct. En général dans un corpus, la note attribuée à la réponse de référence est représentée par plusieurs notes chacune attribuée par un expert. La subjectivité est le problème le plus crucial pour la construction de corpus efficace. A titre d'exemple Mohler (Mohler 2009) a reporté des différences très graves dans la notation de réponses faite lors de la construction d'un corpus. Sur une échelle de notation de 5, certaines différences étaient de 4 points.

Habituellement, pour montrer l'efficacité d'une approche d'évaluation automatique de réponse courte, on utilise Le coefficient de corrélation de Pearson pour montrer la tendance de l'évaluation d'une approche par rapport à une moyenne de qualités humaines (moyennes des notes données par les experts).

Le calcul de similarité entre deux textes représente ainsi le cœur des outils et approches traitant de l'évaluation des réponses courtes. Nous distinguons 2 grandes catégories dans le domaine de la similarité textuelles :

- La similarité entre chaînes de caractères
- La similarité sémantique

En pratique, les outils les plus récents combinent ces deux catégories de similarité. Souvent, le processus global de comparaison entre deux textes commence par l'application de 1 ou plusieurs algorithmes de similarité entre chaîne de caractères. Par la suite c'est la similarité sémantique qui est appliquée

Dans ce qui suit nous présenterons la similarité de chaîne de caractères et les approches qui y sont basées, ensuite les différentes approches de similarité sémantique et nous terminerons par les approches qui combinent les 2 similarités

I.2. Similarité syntaxique ou Similarité entre chaîne de caractères :

Deux types de similarité sont utilisés: similarité exacte de chaîne de caractère et similarité basé sur le calcul d'une distance de similitude. La similarité exacte représente la technique la plus simple utilisée pour évaluer une réponse courte Dans cette technique naïve la réponse de l'étudiant doit correspondre exactement à l'une des réponses de référence afin d'obtenir le score associé. Si aucune correspondance exacte n'est trouvée, la réponse est notée par un 0. Ce type de similarité est réalisé au niveau du caractère

La distance de similarité est un nombre représentant une indication spécifique à l'algorithme concernant la similitude ou de dissemblance entre une réponse de l'élève et une réponse de référence. La notation de la réponse de l'étudiant dépend de cette distance. Cette technique est utilisée intensivement dans de nombreux domaines tels que, la recherche d'information, la traduction automatique de texte. Dans cette technique, la mesure de la distance peut être réalisée soit au niveau des caractères (similarité basé sur le caractère) ou au niveau du terme (similarité des mesures fondées sur le terme-).

I.2.1. La similarité exacte :

Malgré sa simplicité, la technique de similarité exacte est en réalité utilisée par les outils de réponses courtes déployés dans le cadre de plate-forme bien connue de télé-enseignement (LMS pour Learning Management System) tels que Moodle, eFront et Claroline. Ces outils effectuent une simple comparaison de chaînes entre la réponse de l'étudiant et un ensemble de réponse définie par l'enseignant. La réponse de chaque enseignant, spécifié en une seule ligne, est associée à une note allant de 100% à 0%. Dans Moodle, l'enseignant peut préciser si la comparaison doit être sensible à la casse ou non.

Ces outils doivent être utilisés dans la situation où une réponse doit correspondre totalement à une des réponses fournies par l'enseignant afin d'obtenir la note associée à la réponse de l'enseignant, sinon la note est 0. En fait, il existe de nombreux domaines où de tels outils peuvent être efficaces. Dans ces domaines, comme la science religieuse, la poésie, l'écriture de programme informatique et la spécification d'une interaction à l'aide d'un protocole de communication, même une petite erreur ne sont pas acceptées. Malgré la nature exacte de la réponse attendue, les enseignants sont confrontés à de nombreuses difficultés effectivement avec ces outils. A titre d'exemple, dans Moodle outil à réponse courte, un caractère supplémentaire vide peut conduire à une erreur. L'enseignant doit fournir aux étudiants de nombreux conseils afin de les laisser écrire leurs réponses très attentivement.

Dans Moodle, l'outil d'évaluation de réponse courte **shortanswer** offre une option qui tend à réduire l'impact de caractère supplémentaire et même les erreurs d'orthographe. Cependant cette approche risque d'être dangereuse dans certains cas où des réponses fausses sont assimilées à des réponses justes. La technique utilisée est basée sur l'utilisation d'un joker représenté par le caractère asterix (*)

L'astérisque (*) comme joker permet d'indiquer n'importe quel caractère ou chaîne de caractère peut être mise à sa place. Par exemple, l'écriture *pla*ement* correspond à tout mot ou toute proposition commençant par *pla* et se terminant par *ement*. Si on définit une question exigeant deux réponses *carburant* et *oxygène*, on peut limiter le nombre de variantes en écrivant *carburant*oxygène*. Cela acceptera les réponses « *carburant oxygène* », « *carburant, oxygène* », « *carburant; oxygène* », « *carburant et oxygène* », « *carburant & oxygène* », « *carburant/oxygène* », « *carburant and oxygène* ». Toutefois, cette technique de joker acceptera aussi les réponses telles que « *carburant ou oxygène* », « *carburant sans oxygène* » et « *carburant|oxygène* », ce qui pourrait être moins bon.

Voici quelques réponses et notes pour la question « Que consomme une fusée ? ».

1. *oxygène*carburant* - note : 100 %
2. **carburant** - note : 50 %
3. **oxygène** - note : 50 %
4. **air** - note : 40 %
5. *** - note : 0 %

Dans l'outil **Shortanswer** de Moodle, l'ordre des réponses est important. Les réponses sont évaluées de la première à la dernière. Lorsqu'une réponse correspondante est trouvée, le processus s'arrête. Si aucune correspondance n'est trouvée, la note est de 0 et la réaction générale est utilisée. Il est bon de mettre simplement un joker comme **toute dernière réponse** pour que le processus d'évaluation sache quoi faire si aucune correspondance n'est trouvée.

La technique du joker a l'air vraiment utile, mais elle peut être dangereuse et causer des fatalités, que même les développeurs de MOODLE l'ont admis. L'utilisation de '*' peut conduire à accepter des réponses erronées et/ou refuser des réponses plus ou moins proches.

Certains outils permettent plus de flexibilité et de contrôle en spécifiant les réponses à l'aide d'expressions régulières. Grâce aux expressions régulières. L'objectif principal des expressions régulières est de spécifier en une seule écriture un nombre important d'alternatives acceptables et équivalentes sur le plan de la note à affecter. Moodle Regex (Moodle 2015) et WebLas (Bachman 2002) sont deux outils basés sur les expressions régulières. Dans Moodle Regex, la spécification des réponses contenant des expressions régulières est faite manuellement. Une formation sur l'utilisation des expressions régulières est nécessaire pour un enseignant non informaticien. Le gros problème de cette approche est la difficulté de compréhension et mise en œuvre des expressions régulières, parfois même pour des informaticiens. Moodle Reg Ex souffre d'une autre grande difficulté : Il n'est pas compatible avec toutes les versions de Moodle, notamment les plus récentes. A titre d'exemple, la version actuelle de Moodle Reg Ex est valable pour Moodle 2.6 mais pas pour les versions 2.7, 2.8 et 2.9

Dans WebLas (Bachman, 2002), une expression régulière limitée est créée automatiquement pour chaque modèle de réponse après avoir défini, sous le contrôle de l'enseignant, les alternatives acceptables (synonymes) pour chaque élément jugé important dans la réponse. En fait, l'expression régulière de WebLas a le mérite de faciliter la création de multiples réponses de références semblables au lieu de laisser l'enseignant utiliser les expressions régulières lui-même. Selon (Bachman, 2002), les expressions régulières supportées sont très simples car elles sont construites en utilisant uniquement l'opérateur ou (alternatives). Cependant, selon l'étude approfondie faite par (Ziai et al, 2012) WebLas souffre fortement du manque d'une étude d'évaluation sur les bases de données réelles.

Malgré la flexibilité qu'ils apportent, les expressions régulières souffrent d'un autre problème dans le domaine de l'évaluation automatique de réponses courtes : toutes les alternatives ne peuvent pas avoir des notes distinctes. Toute réponse qui satisfait l'expression régulière représentant la réponse de référence, aura la note associée à cette expression régulière

I.2.2. La distance de similarité :

Le calcul de similarité entre chaîne de caractère n'est pas une technique récente. Le Vector Space Model (Salton 75) semble être l'une des premières approches traitant de la similarité de chaînes de caractères. Le VSM a été utilisé pour l'indexation automatique. La métrique de

similarité utilisée par SVM est le coefficient de cosinus, qui mesure l'angle entre un vecteur qui représente un document de référence et un autre vecteur représentant une requête.

Aujourd'hui, un grand nombre important d'algorithmes calculant la distance similarité existent. Plusieurs d'entre eux sont maintenant utilisés dans l'évaluation de réponses courtes.

L'algorithme BLEU (Papineni et al. 2001), conçu à l'origine pour évaluer les systèmes de traduction automatique, est utilisé dans (Perez 2004) pour évaluer les réponses courtes. Perez (2004) a montré que pour un type spécifique de catégorie de question l'algorithme de BLEU atteint de meilleurs résultats que d'autres algorithmes basés sur les mots-clés. L'algorithme de BLEU calcule une métrique basé sur le n-gram (G.Kondrak. 2005) et applique un facteur de pénalité pour pénaliser les textes trop courts. Selon (Perez 2004) la modification du facteur de pénalité peut conduire au résultat plus intéressant surtout quand le métrique uni-gramme (G.Kondrak. 2005) est utilisé.

I.3. Similarité sémantique :

La Similarité sémantique est généralement obtenue soit en se basant sur un corpus, sur une base de connaissance ou les deux. (Gomoa 2013). Le calcul de similarité sémantique basé sur un corpus détermine la similitude entre les mots selon l'information obtenue à partir d'une grande collection de textes qui est utilisé pour la recherche sur la langue. La Similarité basée sur une base de connaissance mesure le degré de similitude entre les mots en utilisant des informations provenant de réseaux sémantiques. WordNet représente le réseau sémantique plus populaire où les noms, verbes, adjectifs et adverbes sont regroupés en ensembles de synonymes cognitifs (synsets), chacun exprimant un concept distinct. Les Synsets sont reliés entre eux par des relations sémantiques et conceptuels-lexicales (Gomoa 2013).

Les techniques du calcul de la similarité sémantique ont été classées par (Mitchell et al. 2002) en trois catégories principales: les techniques statistiques, l'extraction de l'information et le traitement du langage naturel.

La reconnaissance de paraphrase est une approche utilisée par le système C-Rater (Leacock et Chodorow, 2003). Ce dernier a été évalué sur des réponses ayant une moyenne de 43 mots et sur la réponse ayant une moyenne de 15 mots. Dans les deux cas, le pourcentage d'accord entre C-Rater et les experts humains a atteint 84%.

Les techniques d'extraction d'information (IE) soutenues par des techniques de Traitement Automatique de la Langue Naturelle, notamment celles qui traitent les fautes d'orthographe et la segmentation de phrase en noms et verbes, ont été utilisées dans les outils académiques tels qu'AutoMark (Mitchell et al. 2002) et le système de l'Oxford rapportés dans (Pulman, 2005). L'accord signalé entre les experts et la machine peut atteindre 96% pour AutoMark et 84% pour le système de l'Oxford. L'excellent accord AutoMark est effectivement dû au fait que quatre types de la réponse de l'étudiant prévu ont été utilisés: mot unique, valeur unique, une phrase courte explicative et une phrase descriptive. Il est clair que la complexité linguistique pour ce dernier est plus élevée que le troisième qui à son tour est plus complexe que les deux premières. L'accord pour un mot unique et une seule valeur a atteint 100% alors que le maximum obtenu pour le quatrième type de réponse est de 87%.

De nombreuses études ont montré que, contrairement aux techniques de l'IE et des techniques de TALN, techniques statistiques représentées principalement par le « Latent Semantic Analysis » (LSA) (Landauer et Dumais 1997) ne sont pas bien adaptées pour l'évaluation de réponse courte (Dessus et al 2000; Islam 2008). LSA est à la base de plusieurs outils académiques et commerciaux comme Apex Assessor (Dessus et al. 2000], RMT (Méthodes de recherche Tuteur) (Wiemer-Hastings et al. 2004) et l'AIE (Essay Intelligent Assessor (Foltz et al, 1999.).

Nous tenons à remarquer que des études étendues sur les diverses techniques de similarité sémantique ont été reportées dans (Perez, 2007; Ziai et al, 2012) et (Gomoa 2013).

I.4. Evaluation des approches et systèmes actuels :

Les performances déclarées des différents systèmes d'évaluation automatique de réponses courtes qu'ils soient commerciaux ou académiques semblent être très intéressants. Toutefois, ces performances ont été obtenues en réalité dans des environnements spécifiques et favorables aux approches et outils. A titre d'exemples ces environnements sont caractérisés par des aspects très particuliers tels que le type de questions (énumération, justification de choix, définition, description etc..) et la présence de corpus très liés au type de questions. En outre, chaque système vise un cours spécifique et les évaluations des approches et systèmes sont généralement basés sur des réponses écrites correctement et, dans certains cas rares, quelques réponses peuvent contenir des erreurs d'orthographe simples.

Dans la réalité, les environnements favorables ne peuvent pas être trouvés facilement dans la plupart des institutions d'enseignement telles que les universités ou même un département dans une université. Dans certains établissements d'enseignement, comme la plupart des universités algériennes l'environnement est plutôt hostile pour l'évaluation automatique de réponses. L'hostilité est représentée par de nombreux facteurs tels que la grande variété de cours, le nombre des langues utilisées, le niveau bas en langue des étudiants et parfois même d'un nombre important d'enseignants. A titre d'exemple, dans les universités algériennes, les fautes d'orthographe et erreurs homophoniques représentent une situation courante dans les réponses des élèves et même dans certaines réponses fournies par un enseignant on peut trouver ces types d'erreurs.

En raison de leur spécificité (de dépendance de langue, cours spécifique, type spécifique de question, corpus spécifique pour un genre de question) et à l'hostilité de l'environnement (difficulté à fournir corpus, niveau de langue pauvre) les Systèmes d'Evaluation Automatique de Réponses Courtes (SEARC) ne peuvent pas atteindre l'efficacité attendue dans un véritable environnement d'enseignement. De plus, le non support par les systèmes actuels d'un certain nombre d'aspects importants tels que la prise en compte des préférences de l'enseignant, le concept de contenu dynamique dans une réponse courte et les réponses courtes multiples va diminuer l'efficacité de ces systèmes et augmenter la difficulté de leur adoption dans un environnement d'enseignement réel.

A titre d'exemple, aucun SEARC et aucune des approches proposées pour l'évaluation automatique des réponses courtes ne permet de prendre en compte le fait que la réponse est évaluée selon une stratégie d'évaluation qui peut évoluer au cours du processus d'enseignement. La stratégie d'évaluation est décidée par l'enseignant. De plus cette stratégie pourrait être spécifique pour chaque réponse dans un examen en ligne. L'enseignant décide pour chaque question ce qui est important ou non, et quelle est la pénalité à appliquer pour une réponse partielle. C'est l'enseignant qui décide quelles sont les erreurs qui n'ont aucun impact sur une réponse, et celles qui détruisent toute la réponse, même si cette réponse paraît partiellement correctes.

La majeure partie des travaux de recherche se focalisent surtout sur l'amélioration du processus de calcul de similarité sans prendre en compte le fait que, en plus du modèle de réponse, un professeur peut fournir des conseils, des astuces, des contraintes et d'autres informations qui sont capitales pour l'efficacité du processus d'évaluation d'une réponse. D'un point de vue architectural (architecture des logiciels), il apparaît clairement que les

systemes actuels ne donnent pas une attention suffisante aux preferences des enseignants, de leurs orientations et de leurs decisions avant de commencer le processus de comparaison. Les preferences et orientations de l'enseignant peuvent d'un part rendre un SEARC plus independant de son environnement et d'autre part ils peuvent le guider efficacement durant le processus de calcul de similarite.

Solution
proposée

II.1. Introduction générale au système proposé :

Dans le système que nous proposons, un rôle de préparation très important est donné à l'enseignant. Ce rôle sera joué par l'enseignant durant la phase de préparation du système pour l'évaluation automatique d'une réponse. Dans le travail de préparation, l'enseignant indiquera de manière explicite ce qui est important dans une réponse et ce qui n'est pas important. Il pourra aussi indiquer si certaines techniques faisant partie du traitement automatique de la langue naturelle peuvent être utilisées telle que l'exploitation des synonymes.

L'objectif du travail de préparation est l'obtention pour chaque réponse de l'enseignant un corpus dans lequel nous trouverons les réponses probables que l'étudiant pourrait donner. La richesse du corpus dépendra du nombre de directives et indications que l'enseignant aurait donné durant la phase de préparation. Le corpus contient des réponses exactes ou partielles et avec chaque réponse le corpus reporte la note associée. Certaines directives de l'enseignant peuvent aussi être appliquées à la réponse de l'étudiant. A titre d'exemple si la réponse de référence est le mot **été**, et si la réponse de l'étudiant est le mot **étaient**, la directive de pénalisation de l'oubli d'accent produira un corpus comportant les mots **été, éte, eté** et **ete**. Si en plus une directive indique de considérer les réponses composés des 2 premières lettres, c'est-à-dire **ét**, le corpus sera enrichi des mots **ét** et **et**. Ce même processus est appliqué à la réponse de l'étudiant.

Une fois le travail de préparation réalisé par l'enseignant, le système pourra évaluer la réponse de l'étudiant en se basant sur le corpus de référence et la réponse de l'étudiant. Le travail d'évaluation automatique d'une réponse consistera à trouver dans le corpus les réponses les plus similaires. Le facteur de similarité accepté doit être en dessus d'un seuil que l'enseignant fixera. Dans le cas où plusieurs réponses obtiennent un facteur de similarité supérieur au seuil, la réponse retenue sera celle ayant la plus haute note.

II.2. Architecture général du système :

L'architecture générale (Figure 3.1) reflète clairement les 2 phases par laquelle passe l'évaluation automatique de réponse courte.

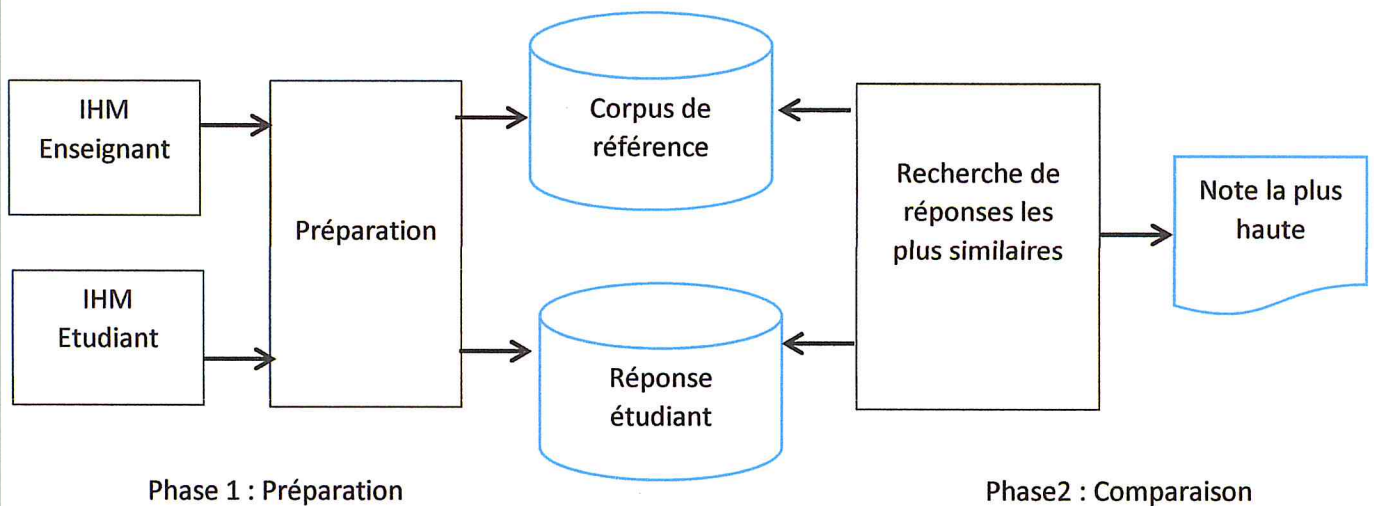


Figure 3. 1 : Architecture générale du système

Dans la première phase, l'enseignant, à travers l'IHM du système va spécifier trois catégories d'informations

1. La question : celle-ci peut contenir que du texte.
2. La (ou les) réponse(s) totalement correcte(s)
3. Les directives et indications permettant de produire le corpus relatifs à chaque réponse. .

Les directives et indications sont spécifiées à travers des contrôleurs. Chaque contrôleur adresse un type particulier de directive pour l'indication et peut offrir divers options pour bien affiner le travail de la directive.

Les contrôleurs sont sollicités pour la production du corpus. L'ordre de mise en œuvre des contrôleurs dépend de la priorité fixé pour chaque contrôleur.

Certain contrôleur (ou directives) s'appliquent seulement au corpus de référence. D'autres s'appliquent uniquement à la réponse de l'étudiant. Plusieurs autres s'appliquent simultanément au corpus et la réponse de l'étudiant. A titre d'exemple la directive « accepter l'absence de mot clés » n'est appliquée que sur le corpus de référence. Les indications comme

« accepter l'absence d'accent », accepter les homophones et accepter l'absence de déterminant s'appliquent au corpus et à la réponse de l'étudiant.

Certaines indications comme la considération des synonymes et des homophones prennent la phase de préparation à une situation de création de dictionnaires locaux. Le contenu d'un dictionnaire local est soit défini manuellement par l'enseignant, soit généré automatiquement sous le contrôle de l'enseignant à partir d'un dictionnaire de domaine. L'enseignant peut ensuite ajuster le dictionnaire généré automatiquement.

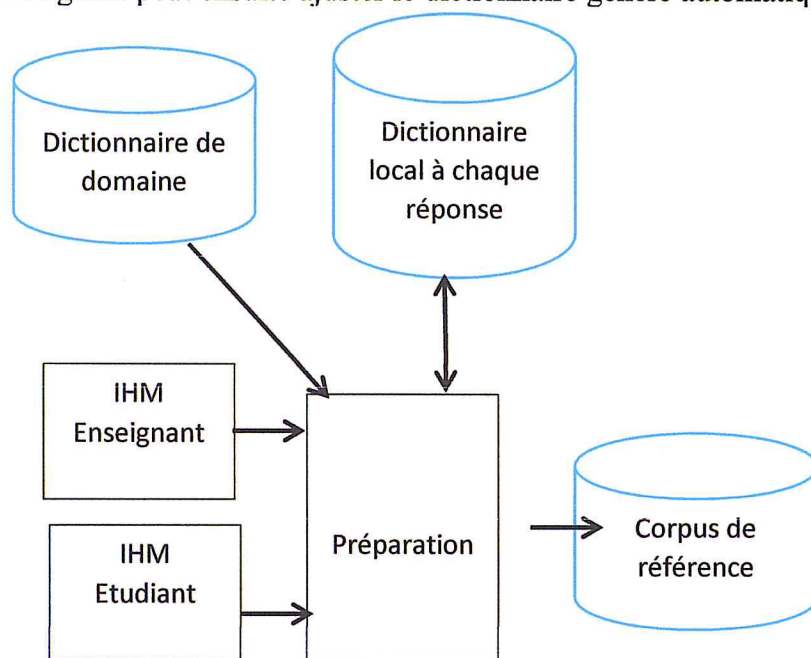


Figure 3. 2 : Création du corpus

La deuxième partie du système qui se charge de la comparaison des réponses se concentre sur l'application d'un ou plusieurs algorithmes de calcul de similarité. La similarité mise en œuvre est une similarité lexicale (ou de chaîne de caractère). Les similarités sémantiques (notamment celles basées sur les synonymes) ont eu lieu durant la première phase qui est prise en charge par la première partie du système (génération du corpus).

Les algorithmes de similarité de texte calculent un taux de correspondance entre deux chaînes de caractères. La correspondance acceptable par la 2^{ème} phase doit être supérieure à un seuil que l'enseignant fixe. Si le nombre de similarités dépassant le seuil est plus grand que 1, le système retient la similarité correspondante à la note la plus élevée. Dans les tests effectués le taux de similarité utilisé est de 97%. Suite aux résultats obtenus, le taux de similarité peut être modifié par l'enseignant et la comparaison peut être à nouveau relancée.

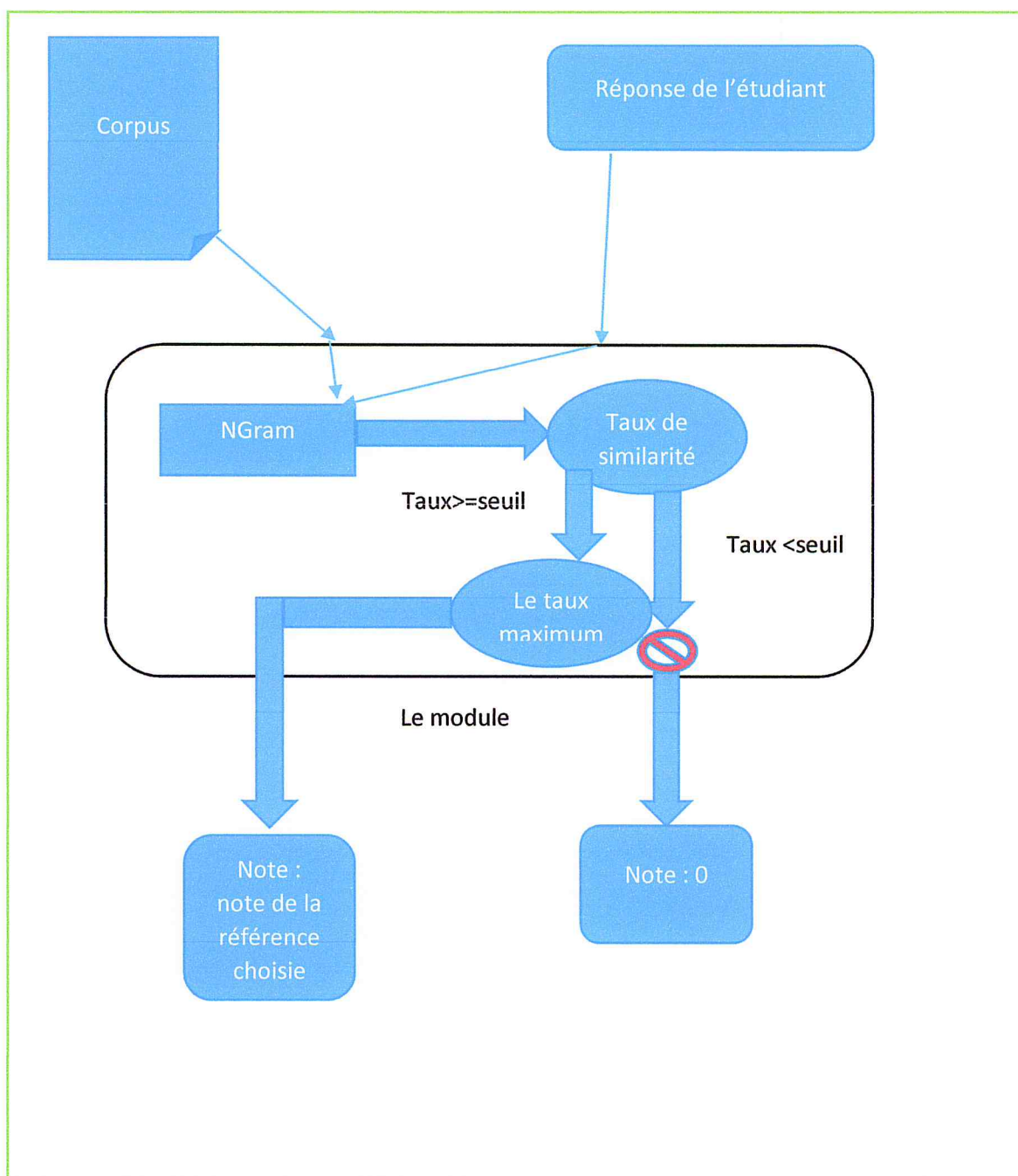


Figure 3. 3 : schéma de la deuxième partie

II.3. Les contrôleurs et directives de la première phase:

Dans la version actuelle, un certain nombre de contrôleurs ont été défini. Chaque contrôleur a un aspect particulier lié à la réponse courte. C'est dans le contexte de chaque contrôleur que l'enseignant définit un certain nombre de directives et fixe certains paramètres que nous appelons souvent indications.

Les directives sont liées à une réponse particulière. Par défaut, lorsque l'enseignant définit plusieurs réponses correctes, les directives définies pour la première réponse seraient normalement automatiquement appliquées aux autres réponses. Cependant, vu que chaque réponse dispose de son ensemble de directives particulières, l'enseignant doit redéfinir pour les autres questions certaines directives et indications

II.3.1. Le contrôleur de choix du seuil et des algorithmes de similarité textuelle :

Ce contrôleur permet de définir d'un seuil de similarité et d'autre part permet de choisir les algorithmes de similarité à mettre en œuvre et leur séquençement si plusieurs doivent être invoqués. Actuellement un seul algorithme est utilisé. Cependant pour des raisons d'extension et amélioration, plusieurs autres algorithmes peuvent être rajoutés. Les algorithmes à rajouter pourraient être des algorithmes de calcul de similarité textuelle ou de calcul de similarité sémantique.

Le seuil de notation automatique : Lorsque le taux de similarité dépasse le seuil de notation automatique, une note est affectée automatiquement à la réponse de l'étudiant. Si plusieurs similarités ont été trouvées supérieures au seuil de notation automatique, ce sera la plus grande note qui sera retenue.

Si aucun des taux de similarité n'atteint le seuil de notation automatique, un 0 est alors attribué automatiquement à la réponse.

II.3.2. Contrôleur de gestion des séparateurs :

Les indications que nous pouvons trouver dans ce contrôleur sont :

- La définition des séparateurs : par défaut il y a deux séparateurs : l'espace et la tabulation

- La suppression des séparateurs
- Réduction du dédoublement des séparateurs

Puisque la liberté d'écriture est donnée à l'enseignant et l'étudiant, les réponses peuvent contenir des espaces inutiles. Cependant, les algorithmes de similarité prennent en compte les espaces. A titre d'exemple, le taux de similarité entre la phrase « **je vais éliminer les blancs** » et la phrase « **je vais éliminer les blancs** » est : de

- **42%** en utilisant les algorithmes NGram et Levenstein.
- **65%** en utilisant l'algorithme de JaroWinkler.

Cependant, si on élimine les espaces, on aura un taux e similarité de 100%. Les espaces jouent un rôle important et ignorer ce détail peut mener au dysfonctionnement de notre outil.

On a deux solutions :

- ✓ Eliminer tous les espaces : une solution facile à implémenter et qui nous évite tant de problèmes, or des fois on aura besoin de séparer quelques mots, et c'est le seul inconvénient de cette solution. Ceci est obtenu par la directive « **supprimer tous les séparateurs** ». Il est clair que dans ce cas l'espace doit être déclaré comme séparateur
- ✓ Eliminer tous les espaces **sauf** l'espace entre deux caractères alphanumériques appartenant à deux mots distincts (s'il y a plusieurs espaces on garde qu'un seul). » les espaces entre les caractères non alphanumériques et les autres seront supprimés. A titre d'exemple « **int ab = 5 ;** » devient « **int ab=5;** » .

II.3.3. Contrôleur de gestion des caractères.

Ce contrôleur gère 2 aspects spécifiques au caractère. Actuellement 2 grands aspects sont considérés dans ce contrôleur : les accents, la sensibilité à la casse et le dédoublement de caractères

II.3.3.1. Option : Les accents :

En réalité cette option est valable que pour quelques langues telles que l'Arabe, le Français et l'Allemand. Dans notre contexte, nous nous sommes intéressés dans un premier temps à la

langue Française. Ce contrôleur doit être réajusté ou tout simplement désactivé selon la langue utilisée

Actuellement, la majorité de nos étudiants ont tendance à omettre les accents ce qui peut mener des problèmes de similarité comme dans le cas des espaces, les algorithmes font la différence entre « e » et « é »,

Exemple :

« j'ai trouvé un élève prêt à générer des questions, ça nous embête, où est la sécurité ? » et « j'ai trouve un eleve pret a generer des questions, ca nous embete, ou est la securite ? ». Les résultats de la similarité fourni par deux algorithmes très connu est la suivante :

86% en utilisant les algorithmes **NGram** et **Levenstein**.

81% en utilisant l'algorithme de **JaroWinkler**.

Bien que 80% est pratiquement un bon taux de similarité, mais il est fort probable qu'il soit inférieur au taux de similarité qui déclenche une notation manuelle ou automatique.

Les options présentes dans ce contrôleur sont :

- Considérer une réponse sans accent : ceci permet de générer une réponse dans le corpus dans laquelle nous ne trouvons aucun accent. Si dans la réponse de l'étudiant il y a au moins un accent présent, une réponse ne contenant aucun accent est ajoutée au corpus étudiant.
- Pénaliser l'oubli de chaque accent : Dans ce cas tous les accents sont pénalisés et supprimés.

II.3.3.2 Option : la casse :

les algorithmes de similarité sont des algorithmes sensible à la casse, ce qui veut dire qu'ils font la différence entre « a » et « A » et comme ça a été déjà mentionné, à cause de la liberté donnée à l'étudiant, on pourra avoir certain majuscule « minuscule » ou inversement. Pour éviter tout cela les deux réponses vont être converties aux minuscules.

II.3.3.3. Option dédoublement de caractère :

Dans la langue française il existe plusieurs mots avec une lettre doublée (à savoir, lettre), plusieurs étudiants se trompe sur ce fait, si c'est évident pour certains mots, ça ne l'est pas pour d'autres mots comme : développer, appeler, appelé, annuler, etc. Pour éviter ce problème

II.3.6. Contrôleur de gestion des synonymes :

Il y a certains mots qui peuvent être remplacés par d'autres mots, le sens peut changer légèrement ou pas du tout, l'enseignant peut ajouter à certains mots de son choix, si l'étudiant utilise ce synonyme, sa réponse sera considérée comme juste, et si l'enseignant définit une pénalité, la note de l'étudiant sera baissée pour l'utilisation de ce synonyme. Ce contrôleur permet de créer pour chaque question un dictionnaire de contenant les synonymes des mots se trouvant dans la réponse de référence. Pour chaque synonyme une question est générée pour le corpus.

II.3.7. Contrôleur de Gestion des homophones :

Le même principe que l'option précédente (synonymes), sauf que cette fois-ci il s'agit des homophones « mots qui se prononcent de la même façon, mais ayant une orthographe différente », comme par exemple « sont » et « son » et « cent » .., l'enseignant peut également définir une pénalité. Grâce à ce contrôleur, car en réalité certains homophone n'ont aucune correspondance avec la sémantique du mot de la réponse de référence. A titre d'exemple, si la question de référence est « cette fois ci le résultat est positif », l'enseignant peut accepter avec pénalité la réponse « cette fois si le résultat est positif » mais ne pourra pas accepter la réponse « sept fois ci le résultat est positif » défini quels sont les homophones à considérer. Ce contrôleur permet de mettre sur pied un dictionnaire de correspondance acceptable entre chaîne de caractère ayant une prononciation très proche

II.3.8. Les fautes d'orthographe :

Les fautes d'orthographe sont fréquentes quand il s'agit d'entrer un texte librement comme le cas de notre outil, il sera judicieux de les éliminer, pour cela on comptera le nombre des fautes ensuite elles seront corrigées, l'enseignant peut définir un facteur de pénalité pour une faute, multiplié par le nombre de fautes, pour diminuer la note de l'étudiant. Si le facteur de pénalité est de 0.5% et que l'étudiant commit 10 fautes, il aura sa note diminuée de $0.5\% \times 10$ soit par 5%.

Une liste des fautes fréquemment commises est établie.

II.3.9. Contrôleur de gestion du contenu dynamique :

Dans certaines réponses, il sera demandé à l'étudiant de proposer lui-même un contenu imprévisible, c'est-à-dire qu'il n'a pas été explicité au niveau de la question de référence. Cette situation est souvent rencontrée lorsque la question laisse à l'étudiant le choix du nom d'un sujet ou l'identificateur d'une variable en langage de programmation. A titre d'exemple si on demande à un étudiant de déclarer une variable `int` dans le langage java que ce soit « `int a ;` », « `int valeur ;` » ou « `int quelquechose 1 ;` » seront tous acceptés.

II.3.10. Les abréviations :

Certains mots peuvent être abrégés, puisque la présence est suffisante, toutefois l'enseignant peut toujours définir une pénalité pour l'utilisation d'une abréviation. .

II.4. La deuxième composante du système :

Cette partie ne se déroule pas au même temps que la première partie de génération du corpus, elle se déroulera quand l'étudiant entre sa réponse, il s'agit de la correction.

Avant l'analyse de la réponse de l'étudiant, le corpus - qui est l'ensemble des réponses possibles - sera généré à partir des options des réponses données par l'enseignant, exemple :

Soit la réponse « `public class A()` » avec `pub` comme abréviation de `public` et `classe` comme homonyme de `class`, on aura les références suivantes :

`public class A()` (réponse originale)

`pub class A()`

`public classe A()`

`pub classe A()`

Chaque référence aura sa propre note, qu'on aura en appliquant les différentes pénalités données par l'enseignant après la génération de l'ensemble des références, la réponse de l'étudiant sera analysée en calculant son taux de similarité avec chaque référence du corpus, on ignorera toute référence dont le taux est inférieure au seuil, la réponse qui a le taux maximum sera choisie comme la réponse fournie et l'étudiant aura la note correspondante.

Si deux références ont le même taux de similarité (maximum) avec la réponse de l'étudiant, la réponse qui a une note plus élevée sera choisie.

Par contre si aucun taux de similarité ne dépasse le seuil, la réponse de l'étudiant sera considérée comme fautive et il aura 0.

II.5. Conclusion :

Pour récapituler, notre solution consiste à générer un ensemble de réponses à partir des réponses originales données par l'enseignant et les options qu'il définira, la réponse de l'étudiant va être comparée (non pas une comparaison brute, mais avec un taux de similarité (distance)) avec l'ensemble des réponses, la réponse la plus correspondante va être choisie en attribuant sa note à l'étudiant.

Cette solution traite la majorité des problèmes rencontrés quand la liberté est donnée à l'étudiant.

Bien que notre outil soit puissant – en toute modestie -, nous ne sommes pas responsables des fautes commises par l'enseignant lors de la génération des réponses.



Implémentation et Tests

III.1. Introduction

Après avoir présenté dans le chapitre précédent notre solution théoriquement, nous allons, dans ce chapitre, détailler l'implémentation de notre solution de correction automatique de réponses courtes. Nous avons dans un premier temps réalisé une application autonome et dans un 2^{ème} cas nous avons réalisé un plugin pour Moodle. La maîtrise de la méthodologie d'écriture d'un plugin pour Moodle est nécessaire pour pouvoir réaliser ce module

On va suivre le plan suivant :

- On va premièrement parler de l'environnement de développement de notre module, les langages et logiciels utilisés, en gros tout ce qui a été utilisé de l'implémentation.
- Ensuite on parlera des algorithmes communs utilisés (dans les deux modules).
- Et finalement on parlera des interfaces des deux modules.
- Clôturer avec une petite conclusion concernant l'implémentation.

III.2. L'environnement de développement:

La liste suivante montre globalement l'environnement de développement utilisé :

- **Système d'exploitation** : Windows 8.1/ Windows 7 Edition intégrale 32 bits.
- **Serveur web**: WAMP (Windows Apache MYSQL PHP).
- **Langages web utilisé** : PHP, HTML/CSS, JavaScript.
- **Langage d'interrogation de Base de données** : MYSQL.
- **Plate-forme** : MOODLE (pour le deuxième module).
- **Navigateur** : Google Chrome.

III.2.1. WAMP :

WampServer (anciennement **WAMP5**) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un *tray icon* (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

Le 1^{er} mai 2014 est sortie la version 2.5. Cette version intègre Apache 2.4.9, MySQL 5.6.17, PHP 5.5.12, PhpMyadmin 4.1.14, SQLBuddy 1.3.3, XDebug 2.2.5.

À ce jour, hormis les add-ons Apache, MySQL et PHP permettant de changer les versions de ces outils, aucun autre add-on n'est disponible (Perl, Webalizer, ZEND Optimizer ou autre).

III.2.2 Le serveur web :

Un serveur HTTP ou démon HTTP ou HTTPd (HTTP daemon) ou (moins précisément) Serveur Web, est un logiciel servant des requêtes respectant le protocole de communication Client-serveur HyperText Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur Web. Le terme « Serveur Web » peut aussi désigner le serveur HTTP (le logiciel) lui-même.

Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le Web et les pages Web sont en pratique toujours servies avec ce protocole. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servies avec d'autres protocoles.

Les serveurs HTTP les plus utilisés sont :

- Apache HTTP Server d'Apache Software Foundation, successeur du NCSA httpd
- Internet Information Services de Microsoft (IIS)
- Sun ONE de Sun Microsystems (anciennement iPlanet de Netscape Communications Corporation)
- Le serveur Web Zeus de Zeus Technology

Le plus populaire est Apache HTTP Server qui sert environ 69% des sites Web en octobre 2005 selon Netcraft.

Historiquement, d'autres serveurs HTTP importants furent CERN httpd, développé par les inventeurs du Web, abandonné le 15 juillet 1996 et NCSA HTTPd, développé au NCSA en même temps que NCSA Mosaic, abandonné mi-1994.

Il existe aussi des serveurs HTTP qui sont des serveurs d'applications capables de faire serveur HTTP, comme Roxen ou Caudium.

III.2.3. Apache HTTP Server:

Le logiciel libre **Apache HTTP Server (Apache)** est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, Common Gateway Interface, Server Side Includes, réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc. Néanmoins, il est à noter que l'existence de nombreux modules Apache complexifie la configuration du serveur web. En effet, les bonnes pratiques recommandent de ne charger que les modules utiles : de nombreuses failles de sécurité affectant uniquement les modules d'Apache sont régulièrement découverts.

Les possibilités de configuration d'Apache sont une fonctionnalité phare. Le principe repose sur une hiérarchie de fichiers de configuration, qui peuvent être gérés indépendamment. Cette caractéristique est notamment utile aux hébergeurs qui peuvent ainsi servir les sites de plusieurs clients à l'aide d'un seul serveur HTTP. Pour les clients, cette fonctionnalité est rendue visible par le fichier `.htaccess`.

Parmi les outils aidant la maintenance d'Apache, les fichiers de log peuvent s'analyser à l'aide de nombreux scripts et logiciels libres tels que, AWStats, Webalizer ou W3Perl. Plusieurs interfaces graphiques facilitent la configuration du serveur.

III.2.4. Langages web :

III.2.4.1. HTML :

L'*Hypertext Markup Language*, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du *Standard Generalized Markup Language* (SGML).

III.2.4.2. CSS

Les **feuilles de style en cascade**, généralement appelées **CSS** de l'anglais *Cascading Style Sheets*, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

L'un des objectifs majeurs des CSS est de permettre la mise en forme hors des documents. Il est par exemple possible de ne décrire que la structure d'un document en HTML, et de décrire toute la présentation dans une feuille de style CSS séparée. Les styles sont appliqués au dernier moment, dans le navigateur web des visiteurs qui consultent le document. Cette séparation fournit un certain nombre de bénéfices, permettant d'améliorer l'accessibilité, de changer plus facilement de présentation, et de réduire la complexité de l'architecture d'un document.

Ainsi, les avantages des feuilles de style sont multiples :

- La structure du document et la présentation peuvent être gérées dans des fichiers séparés ;
- La conception d'un document se fait dans un premier temps sans se soucier de la présentation, ce qui permet d'être plus efficace ;

- Dans le cas d'un site web, la présentation est uniformisée : les documents (pages HTML) font référence aux mêmes feuilles de styles. Cette caractéristique permet de plus une remise en forme rapide de l'aspect visuel ;
- Un même document peut donner le choix entre plusieurs feuilles de style, par exemple une pour l'impression et une pour la lecture à l'écran. Certains navigateurs web permettent au visiteur de choisir un style parmi plusieurs ;
- Le code HTML est considérablement réduit en taille et en complexité, puisqu'il ne contient plus de balises ni d'attributs de présentation.

III.2.4.3. PHP :

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur. PHP peut également générer d'autres formats comme le WML, le SVG, le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP tel qu'IIS. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas toujours une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondants effectuée sur le serveur. La force la plus évidente de ce langage est

qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Il est multiplateforme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (prendre en compte les règles d'arborescences de répertoires qui peuvent changer).

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage.

La version 5.3 a introduit de nombreuses fonctionnalités : les espaces de noms — un élément fondamental de l'élaboration d'extensions, de bibliothèques et de Frameworks structurés, les fonctions anonymes, les fermetures, etc.

Aujourd'hui près de 80% des sites internet utilisent le langage PHP sous ses différentes versions. Plusieurs développeurs PHP responsables de ces sites utilisent en majorité la version 5.4 (38%) dans leurs missions quotidiennes.

Le langage PHP fait l'objet depuis plusieurs années maintenant d'un forum dédié organisé par l'AFUP (l'Association Française des Utilisateurs de PHP), ou experts de la programmation et du milieu se retrouvent pour échanger autour du PHP et de ses développeurs. Ce forum est par ailleurs un moment important pour ces acteurs car il est synonyme du lancement d'une étude dédiée aux développeurs PHP réalisée par l'AFUP et le cabinet de recrutement Agence-E.

III.2.4. JavaScript :

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.

Le langage a été créé en 1995 par Brendan Eich (Brendan Eich étant membre du conseil d'administration de la fondation Mozilla à cette époque) pour le compte de Netscape Communications Corporation. Le langage, actuellement à la version 1.8.2, est une implémentation de la 3^e version de la norme ECMA-262 qui intègre également des éléments inspirés du langage Python. La version 1.8.5 du langage est prévue pour intégrer la 5^e version du standard ECMA³.

III.2.5. Langage d'interrogation de base de données :

III.2.5.1. MYSQL :

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur.

C'est un logiciel libre développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique. Un logiciel qui intègre du code MySQL ou intègre MySQL lors de son installation devra donc être libre ou acquérir une licence payante. Cependant, si la base de données est séparée du logiciel propriétaire qui ne fait qu'utiliser des API tierces (par exemple en C# ou PHP), alors il n'y a pas besoin d'acquérir une licence payante MySQL. Ce type de licence double est utilisé par d'autres produits comme le Framework de développement de logiciels Qt (pour les versions antérieures à la 4.5).

III.2.6. La base de données utilisée :

Nous allons dans cette section la structure de la base de données et ce qu'elle contient comme données

III.2.6.1. Schéma relationnel de la BD :

Admin(user,password) : contient la liste des utilisateurs ayant le profil d'administrateurs du site. Le nom d'utilisateur et le mot de passe sont hachés avec **SHA1**

Branche(nom_branche) : contient toutes les branches concernent nos examens.

Dictionary(id_reference,word,synonyme,penalite) : contient tous les synonymes.

III.3. Les algorithmes relatifs aux directives et indications de l'enseignant :

Dans cette section nous allons présenter quelques algorithmes qui prennent en charges les directives de l'enseignant durant la phase de préparation

III.3.1. Ignorer espaces :

Comme on l'a déjà mentionné, on peut soit éliminer tous les espaces, soit supprimer seulement les espaces entre les caractères alphanumériques et les caractères non-alphanumériques.

L'option de choix sera stockée avec les données de la question, au moment de la correction, on fera l'action correspondante selon le choix de l'enseignant

```

/*****/
function eliminer_espace($id_question,$str){

    $stab=expression($id_question);
    if($stab[0]=='non')
    {
        $str=explode(' ', $str);
        $str=implode(' ', $str);
    }
    else
        $str= eliminer($str,$stab);

    return $str;
}

```

FIGURE 4. 1 : CODE D'ELIMINER ESPACE (1/3)

Si l'enseignant choisira cette option, l'algorithme sera appliqué à la réponse de l'enseignant et de l'étudiant.

Si « \$stab » contient la valeur 'non' ce qui veut dire que tous les blancs vont être éliminés, par contre si l'enseignant choisit de supprimer que les blancs qui sont entre les caractères alphanumériques et les caractères non-alphanumériques l'algorithme suivant sera appliqué :


```

function eliminer2($str) {
    $stab=array();

    $stab[0]="#^[[:alnum:]]#";
    $stab[1]="#[[:alnum:]]$#";
    $stab[2]="#^[[:alnum:]]#";
    $stab[3]="#[[:alnum:]]$#";

    return $this->eliminer($str, $stab);
}

```

FIGURE 4.2 : CODE D'ELIMINER ESPACE (3/3)

```

function eliminer($str, $stab) {

    $str2=explode(" ", $str);

    $final='';

    for($cpt=0;$cpt<count($str2)-1;$cpt++)
    {

        if((preg_match($stab[1], $str2[$cpt]) and !preg_match($stab[2], $str2[$cpt+1])) or
        (!preg_match($stab[3], $str2[$cpt]) and preg_match($stab[0], $str2[$cpt+1])))
            $final=$final.$str2[$cpt];
        else $final=$final.$str2[$cpt].' ';
    }
    $final=$final.$str2[$cpt];
    $str=$final;
    return $str;
}

```

FIGURE 4.3 : CODE D'ELIMINER ESPACE (3/3)

III.3.2. Désactiver la sensibilité à la casse :

Même chose s'applique ici.

```

function negliger_casse($chaine)
{
    return strtolower($chaine);
}

```

FIGURE 4.4 : CODE DE «ELIMINER LA CASSE».

III.3.3. Ignorer les déterminants :

```
function negliger_determinant($chaine)
{
    $determinant=array(" le ", " la ", " les ", "les ", " un ", " une ", " du ", " de ", "dans"
, " la ", " ce ", " cet ", " cette ", " certain ",
" ton ", " son ", " un ", " plusieurs ",
" quelque ", " tout ", " quel ");

    return str_replace($determinant, " ", $chaine);
}
```

FIGURE 4. 5 : CODE DE « ELIMINER DETERMINANTS ».

III.3.4. Ignorer les accents :

```
function negliger_accent($chaine){
    $lettre_a=array("À", "Á", "Â", "Ã", "Ä", "Å", "à", "á", "â", "ã", "ä", "å", "ä", "å");
    $lettre_e=array("È", "É", "Ê", "Ë", "Ê", "Ë", "ê", "é", "ê", "ë", "è", "é", "ê", "ë");
    $lettre_u=array("Û", "Ú", "Û", "Ü", "Û", "Ü", "û", "ú", "û", "ü");
    $lettre_i=array("Î", "Í", "Ï", "Ï", "î", "í", "ï", "ï");
    $lettre_o=array("Ô", "Ó", "Õ", "Ö", "Ö", "ô", "ó", "õ", "ö", "ö");
    $lettre_y=array("ÿ", "Y", "ÿ", "ÿ", "y", "y");
    $lettre_c=array("ç", "Ç");
    $chaine=str_replace($lettre_e, "e", $chaine);
    $chaine=str_replace($lettre_a, "a", $chaine);
    $chaine=str_replace($lettre_u, "u", $chaine);
    $chaine=str_replace($lettre_o, "o", $chaine);
    $chaine=str_replace($lettre_c, "c", $chaine);
    $chaine=str_replace($lettre_y, "y", $chaine);
    $chaine=str_replace($lettre_i, "i", $chaine);
    return $chaine;
}
```

FIGURE 4. 6 : CODE DE « ELIMINER LES ACCENTS ».

III.3.5. Eliminer les doublons :

```
function negliger_double($chaine)
{
    $tab=str_split($chaine);
    $i=0;
    while($i< count($tab)-1)
    {
        if($tab[$i]==$tab[$i+1])
        {
            $tab[$i]='';
            $i++;}
    }
    return implode("", $tab);
}
```

FIGURE 4. 7 : CODE « ELIMINER LES DOUBLONS »

III.3.6. Utiliser Les synonymes :

En ce qui concerne les synonymes, on génère toutes les possibilités pour une référence, c'est-à-dire, si on a la réponse « a b c » (5 points) et a1 ; a2 come synonyme de a et c1 comme synonyme de c, on aura les références suivantes, tous les synonymes avec une pénalité de 10% :

- a b c (original) 100
- a1 b c 90
- a2 b c 90
- a b c1 90
- a1 b c1 80
- a2 b c1 80

Chaque référence aura sa propre pénalité, qui sera appliquée lors de la correction.

Les synonymes sont stockés dans la BD dans la table « dictionary », avant la correction un fichier XML contenant les synonymes et leurs pénalités sera créé.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<body>
  <dictionary>
    <mot>a</mot>
    <synonyme>
      <text>f</text>
      <penalite>50</penalite>
    </synonyme>
  </dictionary>
  <dictionary>
    <mot>b</mot>
    <synonyme>
      <text>g</text>
      <penalite>30</penalite>
    </synonyme>
  </dictionary>
</body>
```

FIGURE 4. 8 : FICHER XML CONTENANT LES SYNONYMES ET LEURS PENALITES.

Pour avoir toutes les possibilités un produit est calculé.

```
function ecrire_synonyme($tab,$id_reference) {
    $cste_xml = "<?xml version='1.0' encoding='UTF-8' ?>";
    $texte = "<body>";
    $sbd=new Sd();
    for($i=0;$i<count($tab);$i++){
        $str=$tab[$i];
        $str=explode(' ', $str);
        for($i=0;$i<count($str);$i++)
        {
            $req=$sbd->select("select * from dictionary where word='".$str[$i]."' and id_reference='".$id_reference."'");
            if($obj = $req->fetch_object()){
                $texte = $texte."<dictionnaire><mot>$obj->word</mot><synonyme><text>$obj->synonym</text><normalite>$obj->normalite</normalite></synonyme>";
                while($obj = $req->fetch_object()){
                    $texte = $texte."<synonyme><text>$obj->synonym</text><normalite>$obj->normalite</normalite></synonyme>";
                }
                $texte = $texte."</dictionnaire>";
            }
        }
    }
    if($texte!="<body>"){
        $texte = $texte."</body>";
        $xml = $cste_xml.$texte;
        $docXML = new DomDocument();
        $docXML->preserveWhiteSpace = false;
        $docXML->formatOutput = true;
        $docXML -> loadXML(utf8_encode($xml));
        $docXML -> save ("xml/synonyme".$id_reference.".xml");
    }
}
```

FIGURE 4. 9 : CODE DE GENERATION DU DICTIONNAIRE XML DE SYNONYME.

```
function get_synonyme($final,$id_reference) {
    $this->ecrire_synonyme($final,$id_reference);

    $taille=count($final);
    $j=0;
    while($j<$taille) {
        $pen=$final[$j+1];
        $str2=explode(' ', $final[$j]);
        for ($i=0;$i<count($str2);$i++)
        {
            $stab=$this->synonyme_mot($str2[$i], $id_reference);
            $final=$this->copy_tab($final, $this->produit($i, $final[$j], $stab, $pen));
        }

        $taille=count($final);
        $j=$j+2;
    }
    return $final;
}
```

FIGURE 4. 10 : CODE D'AJOUT DES SYNONYMES A LA LISTE DES REFERENCES.

Note : la même procédure est utilisée pour les abréviations.

III.3.7. Reconnaître Les Abréviations :

```
function get_abreviation_db($id_reference){
    $stab=array();

    $sbd=new Bd();
    $sreq=$sbd->select("select * from partie where id_reference='".$id_reference."'");
    while($sobj=$sreq->fetch_object()){
        $stab[$sobj->word]=$sobj->partie;
        $stab['note'.$sobj->word]=$sobj->penalite;
    }
    return $stab;
}
}
```

FIGURE 4. 11 : CODE RECUPERATION DES ABREVIATIONS DE LA BD.

```
function add_abreviation_corpus($final, $stab){

    $staille=count($final);
    $sj=0;
    while($sj<$staille){
        $spen=$final[$sj+1];
        $sstr2=explode(' ', $final[$sj]);
        for($si=0;$si<count($sstr2);$si++){
            if (array_key_exists($sstr2[$si], $stab)) {
                $staille=count($final);
                $sstrn=$this->remplacer($si, $final[$sj], $stab[$sstr2[$si]]);
                if(!in_array($sstrn, $final)){
                    $final[$staille]=$sstrn;
                    $final[$staille+1]=$spen-$stab['note'.$sstr2[$si]];
                }
            }
            $sj=$sj+2;
        }
    }
    return $final;
}
}
```

FIGURE 4. 12 : CODE D'AJOUT DES ABREVIATIONS A LA LISTE DES REFERENCES.

III.3.8. Gestion des mots clés :

```
function mot_a_sup($id_reference){
    $stab=array();
    $i=0;
    $bd=new Bd();
    $requete = $bd->select("select * from motasup where id_reference='".$id_reference.'");

    while($objet = $requete->fetch_object()){
        {$stab[$i]=$objet->mot;
        $stab[$i+1]=$objet->pen;
        $i=$i+1;
        }
    }
    return $stab;
}
```

FIGURE 4. 13 : CODE RECUPERATION DES MOTS A SUPPRIMER DE LA BD.

```
function supfinal($final, $os){
    $taille=count($final);
    $j=0;
    while($j<$taille){
        $pen=$final[$j+1];
        $str2=explode(' ', $final[$j]);
        for($i=0; $i<count($str2); $i++){
            if (in_array($str2[$i], $os)){
                $m=count($final);
                $strn=$this->supmot($final[$j], $i);
                if(!in_array($strn, $final)){
                    $final[$m]=$strn;
                    $final[$m+1]=$pen-$this->retpen($str2[$i], $os);}
            }
        }
        $taille=count($final);
        $j=$j+2;
    }
    return $final;
}
```

FIGURE 4. 14 : CODE D'AJOUT DES MOTS A SUPPRIMER A LA LISTE DES REFERENCES.

III.3.9. Correction des fautes d'orthographe :

```

function nombre_faute ($str, $mod, $m1, $m2)
{
    $final = substr_count($str, $m1) - substr_count($mod, $m1);

    if($final > 0)
        return $final;
    else return 0;
}

```

FIGURE 4. 15 : CODE DE CALCUL DE NOMBRE DE FAUTE POUR UN CAS.

Dans cette méthode le nombre des fautes est calculé, concernant deux chaînes la réponse de l'étudiant et la réponse de l'enseignant, et une sous chaîne « m1 » qui sera écrit « m2 »

Exemple : « ç » sera écrit « s ».

```

function nombre_fautes($str, $mod)
{
    $n = 0;
    $l = array('au'=>'q', 'eau'=>'q', 's'=>'c', 'ss'=>'s', 'j'=>'e', 'nh'=>'f',
    'f'=>'nh', 'q'=>'au', 'et'=>'e', 'es'=>'e', 'er'=>'e', 'e'=>'er');
    foreach($l as $key => $val)
    {
        $s = $this->nombre_faute($str, $mod, $key, $val);
        $n += $s;
    }
    return $n;
}

```

FIGURE 4. 16 : CODE DE CALCUL DE NOMBRE DES FAUTES POUR TOUS LES CAS.

III.3.10. Supprimer/Remplacer caractère :

```

function str_src($id_reference, $str)
{
    $bd = new Bd();

    $req = $bd->select("select * from src where id_reference='".$id_reference.'");
    $obj = $req->fetch_object();

    return $this->remplacer_src($str, $obj->src);
}

```

FIGURE 4. 17 : CODE DE RECUPERATION DES CARACTERES A SUPPRIMER/REEMPLACER.

```

function remplacer_src($sch,$modele)
{
    $stab = explode('/', $modele);
    foreach($stab as $s)
    {
        $st =explode(':', $s);

        $sch =  str_replace($st[0], $st[1], $sch);
    }
    return $sch;
}

```

FIGURE 4. 18 : CODE DES CARACTERES A SUPPRIMER/REPLACER.

III.3.11. Les références (Corpus) :

```

function ecrire_reference($stab,$id_reference){
    $scl_xml = "<?xml version='1.0' encoding='UTF-8' ?>";
    $stexte = "<body>";

    for($si=0;$si<count($stab);$si=$si+2)
        $stexte = $stexte."<reference><text>". $stab[$si]. "</text><penalise>". $stab[$si+1]. "</penalise></reference>";
    $stexte = $stexte."</body>";

    $xml = $scl_xml.$stexte;
    $docXML = new DomDocument();
    $docXML->preserveWhiteSpace = false;
    $docXML->formatOutput = true;
    # Créons le fichier XML :
    $docXML -> loadXML(utf8_encode($xml));
    $docXML -> save ("xml/reference".$id_reference.".xml");
}

```

FIGURE 4. 19 : CODE DE L'ECRITURE DES REFERENCES.

Le corpus est représenté par un fichier XML ayant le format ci-dessous.


```

<?xml version="1.0" encoding="ISO-8859-1"?>
<body>
  <reference>
    <text>public static a ; b ; c ; id0 ; id2 ; id0</text>
    <penalite>100</penalite>
  </reference>
  <reference>
    <text>public static a1 ; b ; c ; id0 ; id2 ; id0</text>
    <penalite>90</penalite>
  </reference>
  <reference>
    <text>public static a2 ; b ; c ; id0 ; id2 ; id0</text>
    <penalite>90</penalite>
  </reference>
  <reference>
    <text>public static a ; b1 ; c ; id0 ; id2 ; id0</text>
    <penalite>90</penalite>
  </reference>

```

FIGURE 4. 20 : FICHER XML CONTENANT LES REFERENCES ET LEURS PENALITES.

Un des problèmes rencontrés est les mots avec des accents, le problème était causé par l'encodage des caractères dans le fichier XML l'encodage a été changé, et nous avons utilisé l'encodage UTF-8.

III.3.12. Correction d'une référence :

```

function correct_reference($sid_reference,$repense)
{
  $bd=new Bd();
  $requete2=$bd->select("select * from opt where id_reference='".$sid_reference.'");
  $objet2=$requete2->fetch_object();
  $stab=$this->initialiser($sid_reference,$objet2->dni);
  if($objet2->sn=='true')
  $stab=$this->get_synonyme($stab,$sid_reference);
  if($objet2->prt=='true'){
  $stab2=$this->get_abreviation_db($sid_reference);
  $stab=$this->add_abreviation_corpus($stab,$stab2); }
  if($objet2->mas=='true')
  {
    $stab2=$this->mot_a_sup($sid_reference);
    $stab=$this->supfinal($stab,$stab2);
  }
  $this->ecrire_reference($stab,$sid_reference);
  $grade=$this->correct($sid_reference,$repense,$objet2->na,$objet2->ndt,$objet2->nd,$objet2->nc,
  $objet2->nb,$objet2->dni,$objet2->notedni,$objet2->po,
  $objet2->notepo,$objet2->mc,$objet2->notemc,$objet2->src,$objet2->notena);
  return $grade;
}

```

FIGURE 4. 21 : CODE DE « CORRECTION DE REFERENCE »

On commence par récupérer les options de la référence en question, ensuite on initialisera le corpus avec la référence originale (méthode initialiser).

Ensuite on ajoutera les synonymes au corpus ainsi que les abréviations et les mots à supprimer si ces derniers ont été appliqués.

Après l'ajout des références on corrigera la question avec la méthode « correct » :

```
function correct($sid_reference, $reponse, $na, $ndt, $nd, $nc, $nb, $dyn, $notedyn, $po, $notepo, $mc, $notemc, $src, $notena) {
    $grade=0;
    $tab=array();
    $i=0;
    $xml=simplexml_load_file("$xml/reference/".$sid_reference.".xml") or die("Error: Cannot create object");
    foreach($xml->children() as $syn)
    {
        $pen=0;
        $nbp=0;
        $nna=0;
        $ref=utf8_decode($syn->text);
        $reponse1=$reponse;
        $reponse1=$this->separer($reponse1);
        if($src=='true')
        {
            $ref=$this->str_src($sid_reference,$ref);
            $reponse1=$this->str_src($sid_reference,$reponse1);
        }
        if($nd=='true')
        {
            $ref=$this->negliger_double($ref);
            $reponse1=$this->negliger_double($reponse1);
        }
        if($ndt=='true')
        {
            $ref=$this->negliger_determinant($ref);
            $reponse1=$this->negliger_determinant($reponse1);
        }
    }
}
```

FIGURE 4. 22 : CODE DE « CORRECT » (1/3).

```

if($na=='true')
{
    $nna= $this->cpt_accent($repense1,$ref);
    $nna=$nna*$notena;
    $ref=$this->negliger_accent($ref);
    $repense1=$this->negliger_accent($repense1);
}
if($nc=='true')
{
    $ref=$this->negliger_casse($ref);
    $repense1=$this->negliger_casse($repense1);
}

if($dyn=='true')
{
    if($this->verfier_longueur($ref,$repense1)){
        $pen=$this->verfier_mot($ref,$repense1);
        $pen=$pen*$notedyn;}
    $repense1=$this->changer($ref,$repense1);
}
if($nb=='true')
{
    $repense1= $this->eliminer_espace($id_reference,$repense1);
    $ref= $this->eliminer_espace($id_reference,$ref);
}
if($po=='true')
{
    $nbp= $this->nombre_fautes($repense1,$ref);
    $nbp=$nbp*$notepo;
    $repense1= $this->correct_fautes($repense1);
    $ref= $this->correct_fautes($ref);
}

```

FIGURE 4. 23 : CODE DE « CORRECT » (2/3)

Figure 4.23

```

    $repense1= $this->eliminer2($repense1);
    $ref= $this->eliminer2($ref);
    if($ref==$repense1)
        $grade=max($grade,$syn->penalite-($pen+$nbp+$nna));
    else
        if($mc=='true' and $this->str_mc($id_reference,$repense1))
            $grade=max($grade,$syn->penalite-($pen+$nbp+$notemc));
}
return $grade;
}

```

FIGURE 4. 24 : CODE DE « CORRECT » (3/3)

Les options choisies seront appliquées.

A la fin les taux de similarité des réponses, on négligera les référence qui ont un taux inférieur au seuil, dans ce qui reste on choisira le maximum.

La note sera choisie.

III.3.12. Correction d'une réponse :

```
function correct_question($id_question, $repense, $seuil)
{
    $grade=0;
    $grade2=0;

    $bd=new Bd();
    $n = new NGram();
    $m = new Levenstein();
    $req = $bd->select("select * from reference where id_question='".$id_question.'"");
    while($obj = $req->fetch_object())
    {
        $repense1=$repense;
        $repense1=$this->separer($repense1);
        $requete2 = $bd->select("select * from opt where id_reference='".$obj->id_reference.'"");
        $objet2 = $requete2->fetch_object();
        $grade=$this->maxfloat($grade, $this->correct_reference($obj->id_reference, $repense));
    }

    if($grade<$seuil/100 and $grade2<$seuil/100)
        return 0;
    else
        return $this->maxfloat($grade2, $grade/100);
}
```

FIGURE 4. 25 : CODE DE « CORRECTION DE QUESTION »

Après on fera la même procédure pour toutes les références d'une question, on choisira la valeur maximale.

III.4. L'application autonome de correction de réponses courtes:

Pour le module autonome on a conçu une application web à part :

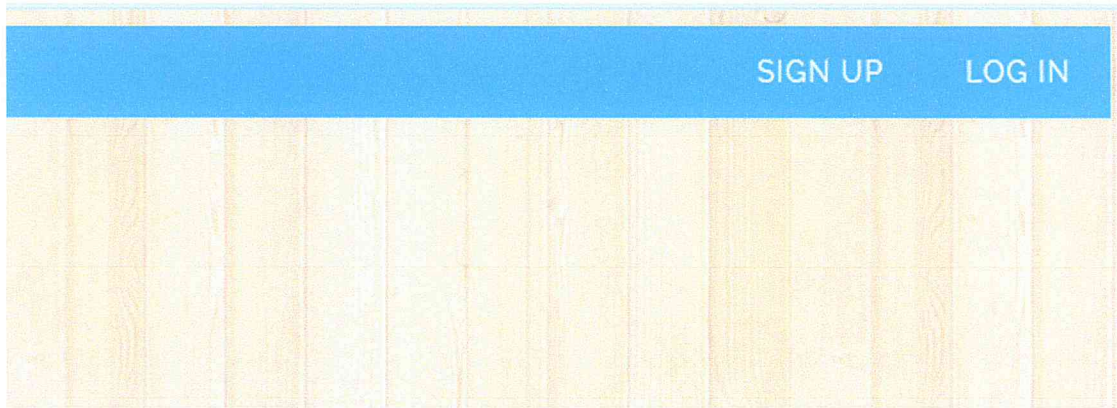


FIGURE 4. 26 : PAGE D'ACCUEIL DE L'APPLICATION

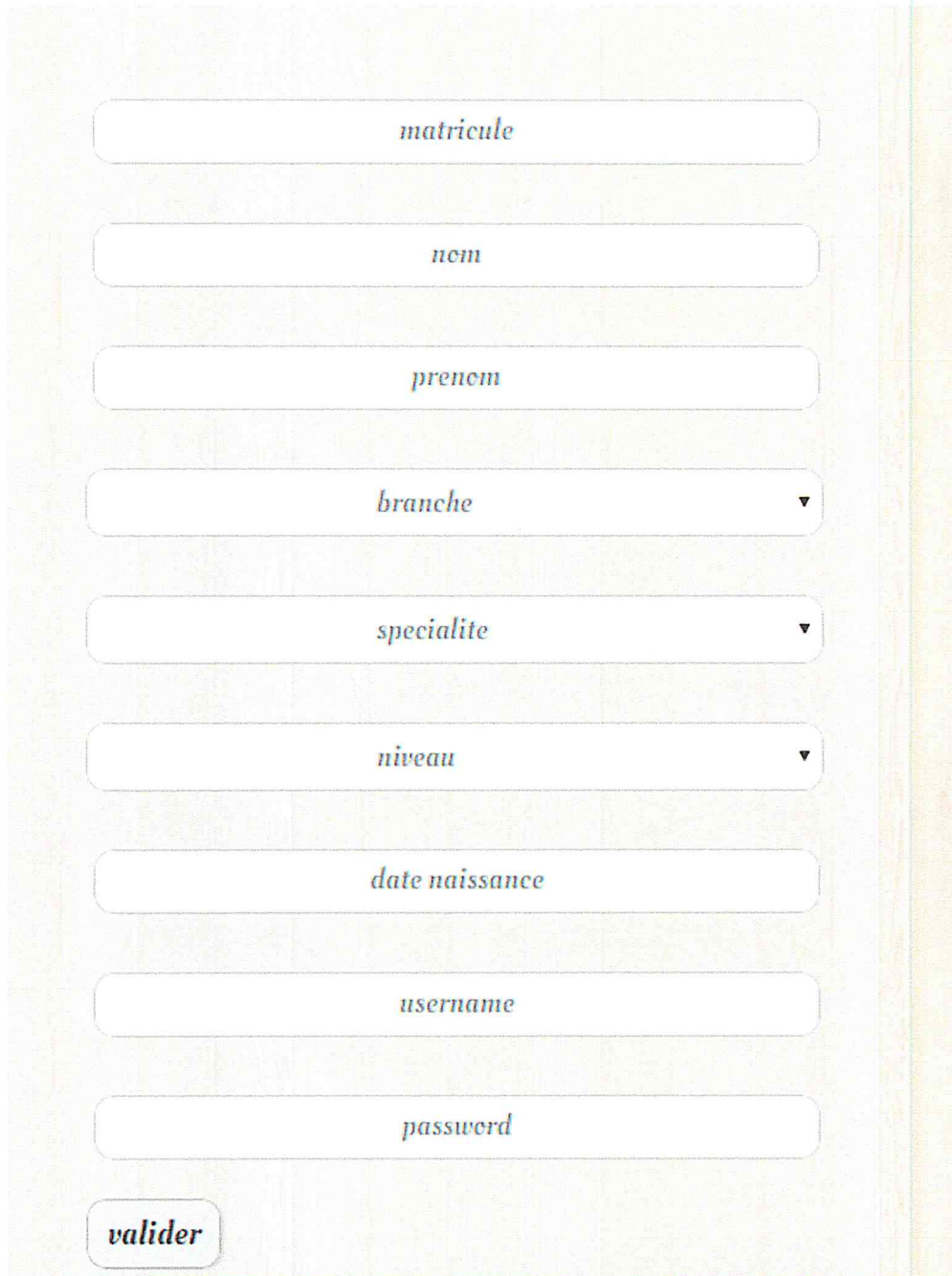
A l'arrivée de l'utilisateur, peut soit se connecter « LOG IN » ou s'inscrire « SIGN UP ».

III.4.1.Inscription :



FIGURE 4. 27 : INTERFACE D'INSCRIPTION (1/3)

L'utilisateur doit choisir s'il s'agit d'un compte d'étudiant ou un compte d'enseignant.



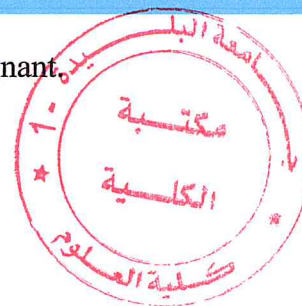
The image shows a registration form with the following fields and controls:

- matricule* (text input)
- nom* (text input)
- prenom* (text input)
- branche* (dropdown menu)
- specialite* (dropdown menu)
- niveau* (dropdown menu)
- date naissance* (text input)
- username* (text input)
- password* (text input)
- valider* (button)

FIGURE 4. 28 : INTERFACE D'INSCRIPTION (3/3)

Après que l'utilisateur eut rempli le formulaire, il valide. Le compte maintenant est créé mais il doit attendre la validation d'un administrateur.

Note : l'inscription de l'étudiant est similaire à celle de l'enseignant.



III.4.2.Connexion :

FIGURE 4. 29 : INTERFACE DE CONNEXION

L'utilisateur entre ses données et se connecte évidemment.

III.4.3.L'enseignant :

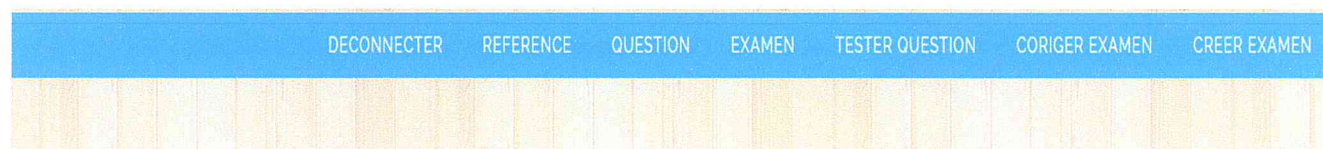


FIGURE 4. 30 : INTERFACE DE L'ENSEIGNANT « MENU D'OUTILS »

L'enseignant peut : Créer un examen, Corriger un examen déjà créé et passé par les étudiants, Tester un examen, éditer un examen, une question d'un examen ou une référence d'une question, sans oublier la déconnection.

III.4.3.1. Création d'examen :



Examen

nom de l'examen

nombre de questions

date de l'examen

heure de l'examen

branche ▼

specialite ▼

niveau ▼

valider

FIGURE 4. 31 : INTERFACE DE CREATION D'UN EXAMEN

III.4.3.2. Edition des questions :

Après le choix du nombre des questions, l'enseignant est amené à générer ses questions grâce à l'interface suivante (la page suivante) :

L'enseignant doit :

- Indiquer le texte de la question, la note et le seuil (cf. Chapitre III, page 11).

- Ajouter la première référence avec ses options
- Ajouter d'autres références en cliquant sur « Add ».

On va maintenant détailler sur les options.

question

note

seuil

reference

100

dynamique synonyme

add

Négliger les blancs: non [ajouter mot](#)

mot dynamique: non

faute: oui

mot clé: non [ajouter mot](#)

Négliger les Accents: oui

synonyme/homonyme: non

Négliger les Doublements: non

Négliger les déterminants: oui

Négliger la casse: oui

supprimer et remplacer caractères: non [ajouter](#)

FIGURE 4.32 : INTERFACE DE CREATION D'UNE QUESTION (1/2).

absence de mot : non [ajouter mot](#)

abréviation : non [ajouter](#)

reference0 [ajouter](#)

valider

FIGURE 4.33 : INTERFACE DE CREATION D'UNE QUESTION (2/2).

Chaque option a un bouton, avec 2 états : oui ou non (voir **figure 4.32**).

Certaines options ont une pénalité.

III.4.3.2.1. Le contenu dynamique :

L'enseignant choisit le mot dynamique puis clique sur le bouton « dynamique » pour l'ajouter, il pourra ensuite en créer un autre.

Il peut également choisir les mots dynamiques qui peuvent être identiques, en tapant les mots dans les « mot 1 » et « mot 2 » respectivement. Quand l'ajout est terminé, l'enseignant clique sur « terminer » pour finaliser, l'interface suivante :



FIGURE 4. 34 : INTERFACE DES MOTS DYNAMIQUES.

III.4.3.2.2. Synonymes :

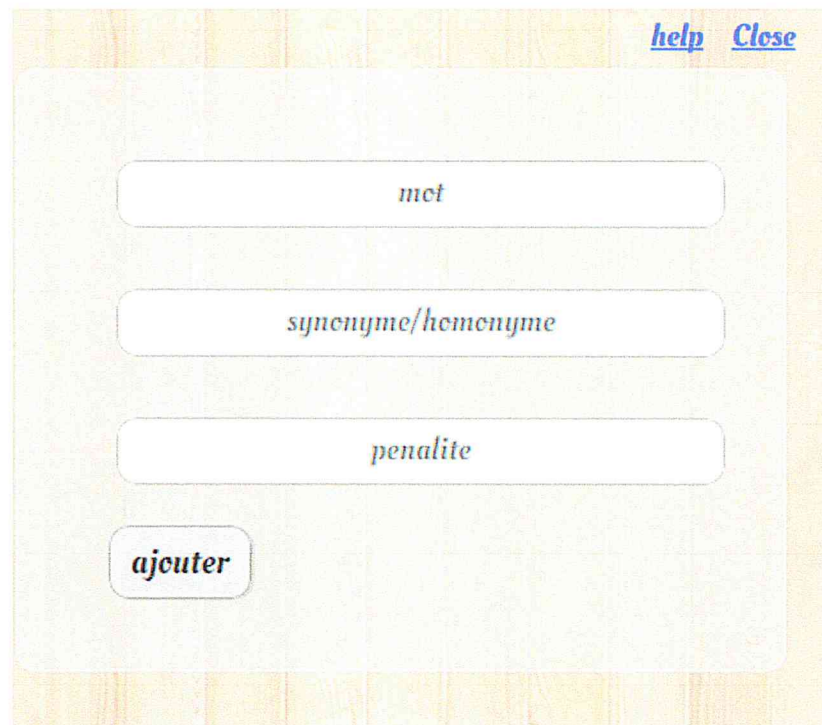


FIGURE 4. 35 : INTERFACE DES SYNONYMES.

Pour ajouter un synonyme, l'enseignant doit définir à quel mot appartient ce synonyme, le synonyme ou éventuellement l'homonyme et finalement la pénalité appliquée si l'étudiant utilise ce synonyme.

III.4.3.2.3. Absence des mots :

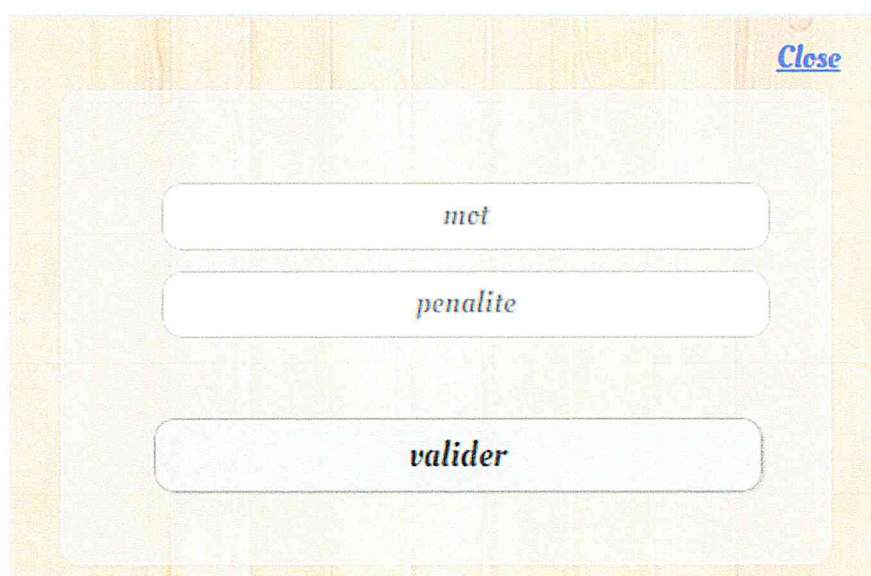


FIGURE 4. 36 : INTERFACE D'ABSENCE DES MOTS

L'enseignant choisit le mot et la pénalité à appliquer en cas de l'absence du mot.

III.4.3.2.4. Mots Clé :



FIGURE 4. 37 : INTERFACE DES MOTS CLE

L'enseignant ajoute les mots clé avec le bouton « ajouter » un par un, et clique sur « terminer » pour finaliser.

III.4.3.2.5. Négliger les blancs :

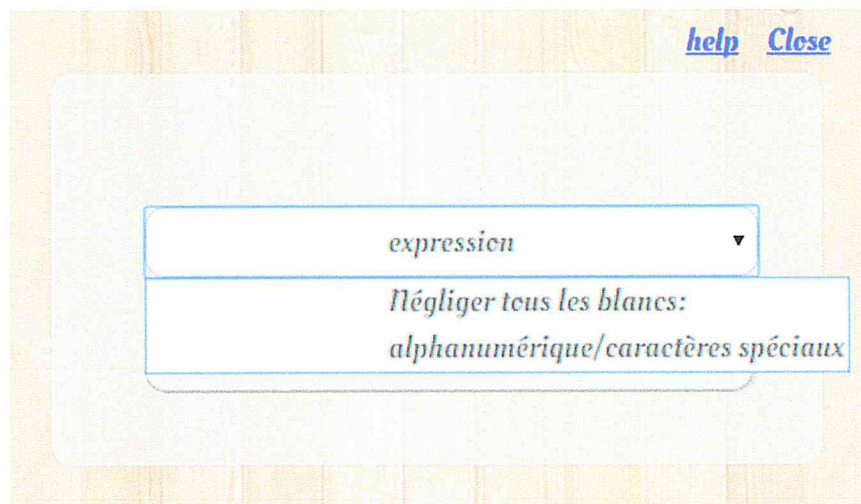


FIGURE 4. 38: INTERFACE DE « NEGLIGER LES BLANC ».

Comme on peut voir, l'enseignant doit choisir s'il néglige tous les blancs ou juste entre les lettres comme on a déjà mentionné.

III.4.3.2.6. Abréviation :

Procédure identique à celle du synonyme.



FIGURE 4. 39 : INTERFACE DES ABREVIATIONS.

III.4.3.2.7. Remplacer/supprimer des caractères :



FIGURE 4. 40: INTERFACE DE « REMPLACER/SUPPRIMER CARACTERE ».

Interface évidente, supprimer ou remplacer «caractere » par « caractere2».

III.4.3.2.8.Choix de la référence :

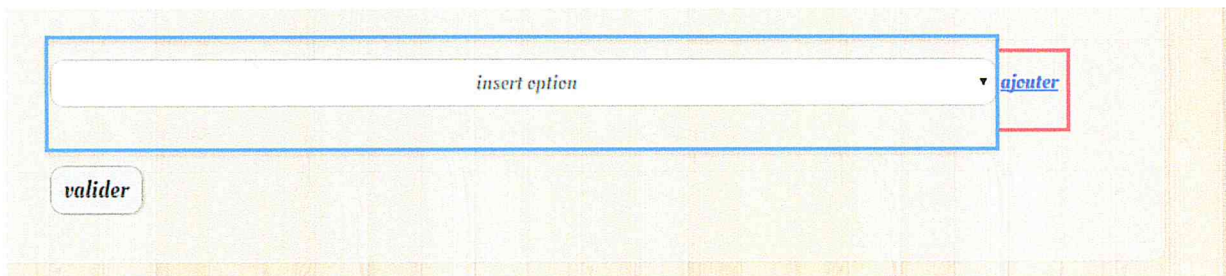


FIGURE 4. 41 : CHOIX DES OPTIONS DE LA REFERENCE (1/2).

Après que l'enseignant eut choisi les options il peut cliquer sur « ajouter » (encadré en rouge) pour définir les options pour la référence choisi dans la liste déroulante (encadré en bleu).

Avec la liste (encerclé en vert dans la figure suivante) il peut choisir la référence à laquelle il veut appliquer les options (s'il en a beaucoup).

FIGURE 4. 42 : CHOIX DES OPTIONS DE LA REFERENCE (2/2).

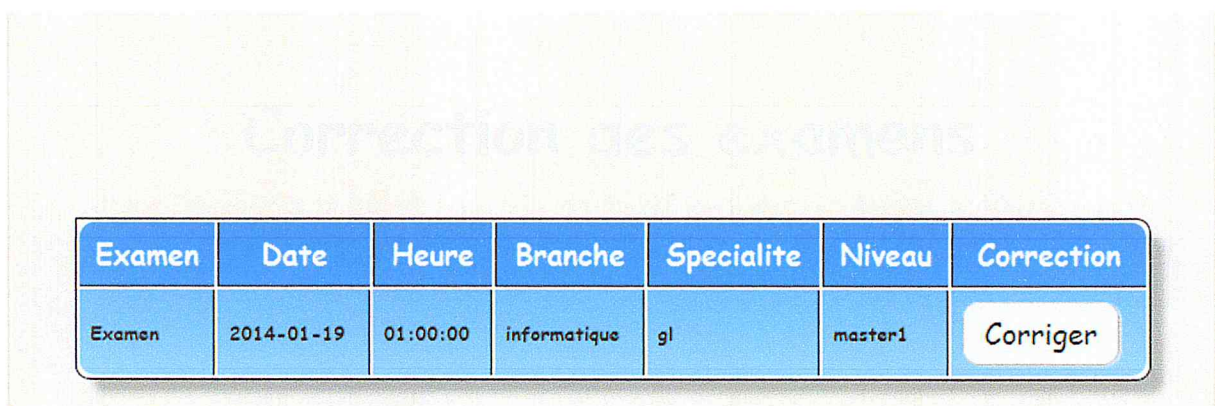
Après que l'enseignant eut terminé de définir les options pour chaque référence, il peut finaliser la question en cliquant sur « valider ».

La même procédure est à refaire pour les autres questions qui restent. Quand c'est terminé l'enseignant est dirigé à la page d'accueil.

III.4.3.2.9 Remarque :

Pour reconnaître les mots, l'enseignant choisit d'appliquer la séparation ou non, ce qui veut dire que les caractères non-alphanumériques seront séparés des caractères alphanumériques, parce exemple « int a=5; » sera « int a = 5 ; », par contre ce n'est pas toujours le cas, dans le cas d'une adresse IP « 192.168.1.1 » il est nécessaire de ne pas séparer la chaîne, ainsi l'enseignant a le choix d'appliquer la méthode de séparation ou non, mais l'enseignant doit – hautement recommandé- séparer lui-même les mots par un espace, ainsi l'outil donnera des résultats efficaces et évitera des erreurs.

III.4.3.3. Correction d'examen :



Examen	Date	Heure	Branche	Specialite	Niveau	Correction
Examen	2014-01-19	01:00:00	informatique	gl	master1	Corriger

FIGURE 4. 43 : INTERFACE DE CORRECTION DES EXAMENS.

L'enseignant peut corriger les examens qu'il a créés, dès que les étudiants l'ont passé.

III.4.3.4. Edition des examens/questions/références :



choix examen

examen ▼

valider

FIGURE 4. 44 : INTERFACE DE CHOIX DE L'EXAMEN.

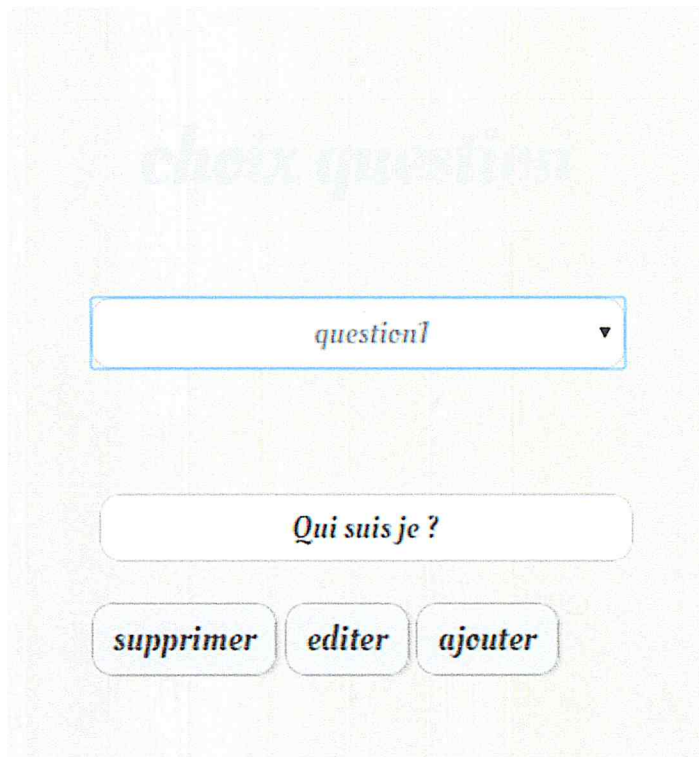


FIGURE 4. 45 : INTERFACE DE CHOIX DE QUESTION.

III.4.3.4.1. Supprimer une référence :

On peut supprimer une référence d'une question existante.

Reference	Note	Supprimer
edg	100	<i>supprimer</i>
dfggh	100	<i>supprimer</i>
dfh	100	<i>supprimer</i>

FIGURE 4. 46 : CHOIX DE LA REFERENCE A SUPPRIMER.

III.4.3.4.2. Editer une référence :

Modifier une référence existante.



FIGURE 4. 47 : CHOIX DE LA REFERENCE A MODIFIER.

III.4.3.4.3. Ajouter une référence:

Ajouter une référence à la question choisie.

L'interface est le même que dans l'ajout d'une référence montré dans la (Figure 4.32/4.33, page).

III.4.3.4.4 Edition d'une question :

The screenshot shows a web interface for editing a question. At the top, there is a dropdown menu with the text 'question1' and a small downward arrow. Below this is a text input field containing the question text 'Qui suis je?'. At the bottom of the interface is a button labeled 'valider'.

FIGURE 4. 48 : INTERFACE D'EDITION D'UNE QUESTION.

Pour l'édition d'une question l'enseignant suit la même procédure pour choisir l'examen que l'exemple précédant. Après il peut soit modifier une question (Figure précédente), supprimer une question de l'examen choisi :

Question	Note	Supprimer
Qui suis je ?	5	Supprimer

FIGURE 4. 49 : INTERFACE DE SUPPRESSION DES QUESTIONS.

Finalement il peut ajouter une question à l'examen choisi.

Et pour lui rajouter des références il doit suivre les procédures montrées ci-dessous.

Ajout d'une question

C'est quci le?

5

70

valider

FIGURE 4. 50 : INTERFACE D'AJOUT D'UNE QUESTION.

Et finalement pour l'édition des examens, la même procédure :

Il peut choisir l'examen à supprimer parmi une liste, à condition que l'examen soit créé par cet enseignant.

Suppression des examens

Examen	Date	Supprimer
Examen	2014-01-19	Supprimer
dd	2015-05-23	Supprimer
test	2015-06-03	Supprimer

FIGURE 4. 51 : INTERFACE DE SUPPRESSION DES EXAMENS.

Il peut aussi modifier les données d'un examen déjà créé comme suit :

Modifier un examen

Examen

2014-01-19

01:00:00

branche ▼

specialite ▼

niveau ▼

valider

FIGURE 4. 52 : INTERFACE DE MODIFICATION D'UN EXAMEN.

III.4.4.Administration :

L'administrateur peut valider des étudiants et des enseignants qui viennent de s'inscrire.

valider des Etudiants à valider

matricule	nome	prenom	valider
0913D030215	B	D	correct

FIGURE 4. 53 : INTERFACE DE VALIDATION DES ETUDIANTS (ENSEIGNANTS).

La même interface existe pour la validation des enseignants.

III.4.5. Passer un examen :

Pour qu'il passe un examen l'étudiant doit être présent entre l'heure du début de l'examen et la fin de l'examen (soit 1h30 plus tard).

Dès qu'il connecte il peut voir l'examen à passer maintenant, bien entendu l'examen qui concerne sa branche et sa spécialité et la date du jour.

Nom de l'examen

question : Question 1?

question : Question 2?

envoyer

FIGURE 4. 54 : INTERFACE DE L'EXAMEN POUR L'ETUDIANT

Comme on peut voir, l'étudiant répond à toutes les questions qu'il peut y répondre et clique sur envoyer pour enregistrer sa réponse, l'enseignant pourra ensuite corriger l'examen comme montré auparavant.

III.4.6. Se déconnecter :

Tous les utilisateurs peuvent se déconnecter évidemment

DECONNECTER **PASSER EXAMEN**

III.5. Le module MOODLE :

Pour l'ajout un module dans une plateforme MOODLE, ce dernier propose un module de base pour pouvoir en créer un nouveau.

On a créé l'interface à l'aide des méthodes fournies par l'API Form de MOODLE.

III.5.1. Introduction à l'API de création de formulaire (formslib) :

MOODLE fournit une puissante bibliothèque de création de formulaire, souple et facile à utiliser.

Normalement on doit créer une nouvelle classe pour notre formulaire qui héritera de « formslib.php », mais comme MOODLE fournit un modèle de base pour un type de question on évitera tout cela.

On peut directement commencer à définir notre formulaire dans le fichier « edit_nomtype_form.php » pour notre cas : « edit_testd_form.php »

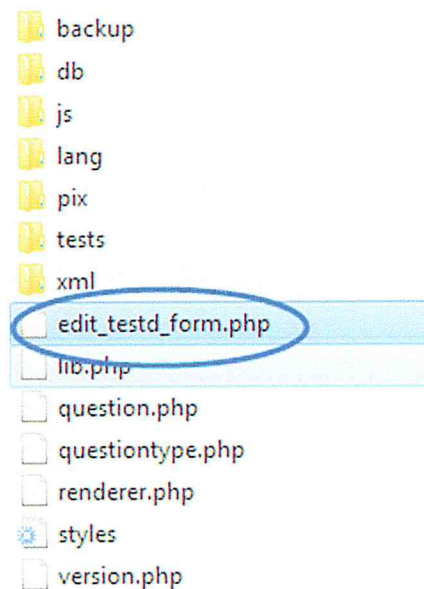


FIGURE 4. 55 : « EDIT_TESTD_FORM.PHP »

Dans ce fichier PHP, on cherche la classe `qtype_testd_edit_form` spécifiquement la méthode « `definition_inner` » pour définir les éléments qui vont être dans l'interface d'ajout d'une question (voir les figures de l'interface MOODLE).

Note : cette méthode permet de définir les éléments spécifiques à notre type « testd », les autres éléments communs aux autres types sont définis ailleurs dans MOODLE.

```
|class qtype_testd_edit_form extends question_edit_form {
|     protected function definition_inner($mform) {
```

FIGURE 4.56 : METHODE DE DEFINITION DU FORMULAIRE

Pour ajouter un élément on utilise la méthode `$mform->addElement()` :

Dans ce qui va suivre, on va voir comment on pourra ajouter chaque type d'élément :

III.5.1.1. Header :

```
$mform->addElement('header', 'options', get_string('option', 'qtype_testd'));
```

FIGURE 4.57 : AJOUTER UN HEADER

Le premier argument indique el type d'élément à ajouter, ici 'header'.

Le deuxième est l'id d'élément dans le code HTML

Le troisième est le nom affiché à côté de l'élément sur la page (on parle de la méthode `getstring()` à la fin)

III.5.1.2. Textfield :

```
$mform->addElement('text', 'seuil', get_string('seuil', 'qtype_testd'), '');
```

FIGURE 4.58 : AJOUTER UN TEXTFIELD

Même chose à dire que le précédent, le quatrième paramètre définit la valeur par défaut du textfield.

III.5.1.3. Checkbox :

Mêem procédure, 'checkbox' comme premier argument.

III.5.1.4. Liste déroulante :

```
$menu2 = array(
    get_string('orb', 'qtype_testd'),
    get_string('aca', 'qtype_testd')
);
$smform->addElement('select', 'testtest',
    get_string('esp', 'qtype_testd'), $menu2);
```

FIGURE 4. 59 : AJOUTER UNE LISTE DEROULANTE

Le tableau menu2 comporte les éléments de la liste déroulante(le dernier argument de la méthode).

III.5.1.5. Bouton :

```
$smform->addElement('button', 'intro', get_string('valider', 'qtype_testd'),'');
```

FIGURE 4. 60 : AJOUTER UN BOUTON

III.5.1.6. La méthode get_string() :

Cette méthode renvoie le texte fourni en paramètre qui est stocké dans le dossier lang :

Cette méthode est pratique pour le multi langues

```
$string['nd'] = 'Négliger les doubles';
$string['ndr'] = 'Négliger les déterminants';
$string['nc'] = 'Négliger la Casse';
$string['src'] = 'Supprimer/Remplacer caractère';
$string['sn'] = 'Synonyme/homonyme';
$string['na'] = 'Négliger les accents';
$string['prt'] = 'abréviation';
$string['dni'] = 'Dynamique';
```

FIGURE 4. 61 : LE FICHIER « QTYPE_TESTD.PHP »

Comme on peut voir, get_string('nd', 'qtype_testd') retournera la chaîne « Négliger les doubles » et ainsi de suite.

III.5.1.7. Autres méthodes :

- **Fermer un header :**

Il existe deux méthodes pour fermer un header, ouvrir un autre header ou utiliser la méthode suivante :

```
$mform->closeHeaderBefore('mas');
```

FIGURE 4. 62 : FERMER UN HEADER

Note : tous ces éléments seront montrés dans la partie des interfaces.

III.5.2. La base de données MOODLE :

MOODLE contient une base de données très riche, pour ajouter des tables spécifiques au module ajouté, on doit modifier un fichier fourni avec le module de base qui se trouve dans le dossier « db », le fichier « install » est un fichier XML, donc il contient des balises :

- la balise <TABLE> pour créer une table qui peut contenir deux partie
 - ✓ <FIELDS> qui contient des <FIELD> chaque field représente un champ de la table
 - ✓ <KEYS> la définition des mots clé

```
<TABLE NAME="qtype_testd_options" COMMENT="Options for short answer questions">
  <FIELDS>
    <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" SEQUENCE="true"/>
    <FIELD NAME="questionid" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false" COMMENT="Forei
    <FIELD NAME="seuil" TYPE="int" LENGTH="2" NOTNULL="true" DEFAULT="0" SEQUENCE="false" COMMENT="Whether a

  </FIELDS>
  <KEYS>
    <KEY NAME="primary" TYPE="primary" FIELDS="id"/>
    <KEY NAME="questionid" TYPE="foreign-unique" FIELDS="questionid" REFTABLE="question" REFFIELDS="id"/>
  </KEYS>
```

FIGURE 4. 63 : EXEMPLE D'UNE TABLE DANS LE FICHIER INSTALL.XML.

Le fichier peut être généré automatiquement avec des outils comme XMLDB.

III.5.3. Remarques :

Pour finaliser l'installation du plug-in voici quelques étapes à suivre :

- Le module de base propose un nom anonyme, les fichiers qui doivent comporter le nom du module doivent voir leur nom modifié.

- Tous les fichiers PHP qui contiennent le nom du module (pas leur nom, mais le contenu) doivent être modifiés, pour cela on ouvrira tous les fichiers PHP du module et on utilisera un module tel que notepad++ pour remplacer le nom existant par notre nom « testd ».
- Le dossier final sera copié dans le dossier suivant : « moodle/question/type », dans la connexion suivante avec un compte admin, on verra une notification d'un nouveau module.
- Après la validation le module sera prêt à l'emploi.

III.5.4. Correction d'une question :

Les algorithmes déjà vu dans la section 2 de ce chapitre seront utilisés dans MOODLE.

La correction sera dans la méthode « grade_reponse » dans le fichier « question.php ».

III.5.5. Interfaces du module de MOODLE :

Note : Dans ce qui va suivre on ne détaillera pas ce qui a été expliqué.

Lors de l'ajout d'une question on peut remarquer la présence de notre plugin « testd »



FIGURE 4. 64 : CHOIX DU TYPE D'UNE QUESTION SOUS MOODLE.

On arrive après à l'interface d'ajout de la question, où on est amené à entrer les différents données, que ce soit des données utilisées par MOODLE, ou par notre module.

Parmi eux on compte : le nom de la question, le texte de la question et la note, ainsi que les options (qui vont être détaillées), sans oublier la/les réponse(s) avec au moins une réponse à 100%.

Nom de question*

Texte de la question*

☰ ↕ B I ☰ ☰ 🔗 🔗 🖼️ 📄

Note par défaut*

FIGURE 4. 65 : INTERFACE D'AJOUT D'UNE QUESTION SOUS MOODLE (1/13).

seuil

Négliger les blancs

Pénalité

► Options

FIGURE 4. 66 : INTERFACE D'AJOUT D'UNE QUESTION SOUS MOODLE (2/13).

Le premier input concerne le seuil, la case à cocher sert à indiquer si les blancs seront négligés ou non, quant aux options, ça concerne les options de négliger les blancs (figure suivante).

Options

espace

Négliger tous les blancs	▼
Négliger tous les blancs	
Alphanémurique/Caractère spéciaux	

FIGURE 4. 67 : INTERFACE D'AJOUT D'UNE QUESTION SOUS MOODLE (3/13)

Rien de particulier, on a déjà parlé sur ça dans l'interface du module autonome.

Absence du mot

Options

Mot

Pénalité

0

Valider

FIGURE 4. 68 : INTERFACE D'AJOUT D'UNE QUESTION SOUS MOODLE (4/13)

Négliger les accents

Pénalité

Négliger les déterminants

Négliger les doubles

Négliger la Casse

Supprimer/Remplacer caractère

FIGURE 4. 69 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (5/13) (NEGLIGER).

Supprimer/Remplacer caractère

▼ Options

Mot

Pénalité

Valider

FIGURE 4. 70 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (6/13)

▼ Options

Mot

Synonyme/homonyme

Pénalité

Valider

FIGURE 4. 71 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (7/13)

partie

Options

Mot

partie

Pénalité

Valider

**FIGURE 4. 72 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (8/13)
(ABBREVIATIONS).**

Dynamique

Pénalité

Options

Mot

Valider

Mot 1

Mot 2

Valider

FIGURE 4. 73 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (9/13)

Fautes d'orthographe

Pénalité 0

mot clé

Pénalité 0

▼ Options

Mot 1

Valider




FIGURE 4. 74 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (10/13) (FAUTES D’ORTHOGRAPHES).

reference

Valider

FIGURE 4. 75 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (11/13) (ENREGISTRER LA REFERENCE).

Pour enregistrer la référence il suffit de choisir le numéro de la référence (encadré en rouge) et cliquer sur « valider » (encadré en bleu).

Answer 1

Note 100% ▼

FIGURE 4. 76: INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (12/13) (AJOUTER UNE REPONSE).

Plusieurs réponses peuvent être ajoutées, la première réponse doit – obligatoirement- avoir une note de 100%.



**FIGURE 4. 77 : INTERFACE D’AJOUT D’UNE QUESTION SOUS MOODLE (13/13)
(ENREGISTRER LA QUESTION).**

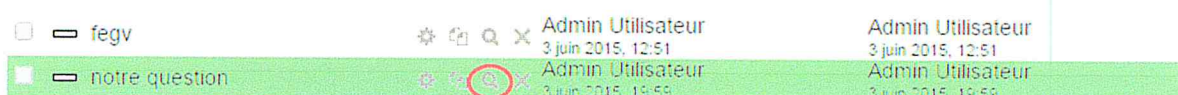


FIGURE 4. 78 : INTERFACE DE LA LISTE DES QUESTIONS.

On peut tester une question en cliquant sur le bouton encerclé en rouge :

Prévisualisation de la question : notre question



FIGURE 4. 79 : INTERFACE DE LA CORRECTION D’UNE QUESTION.

III.5.6. Difficultés rencontrés Moodle :

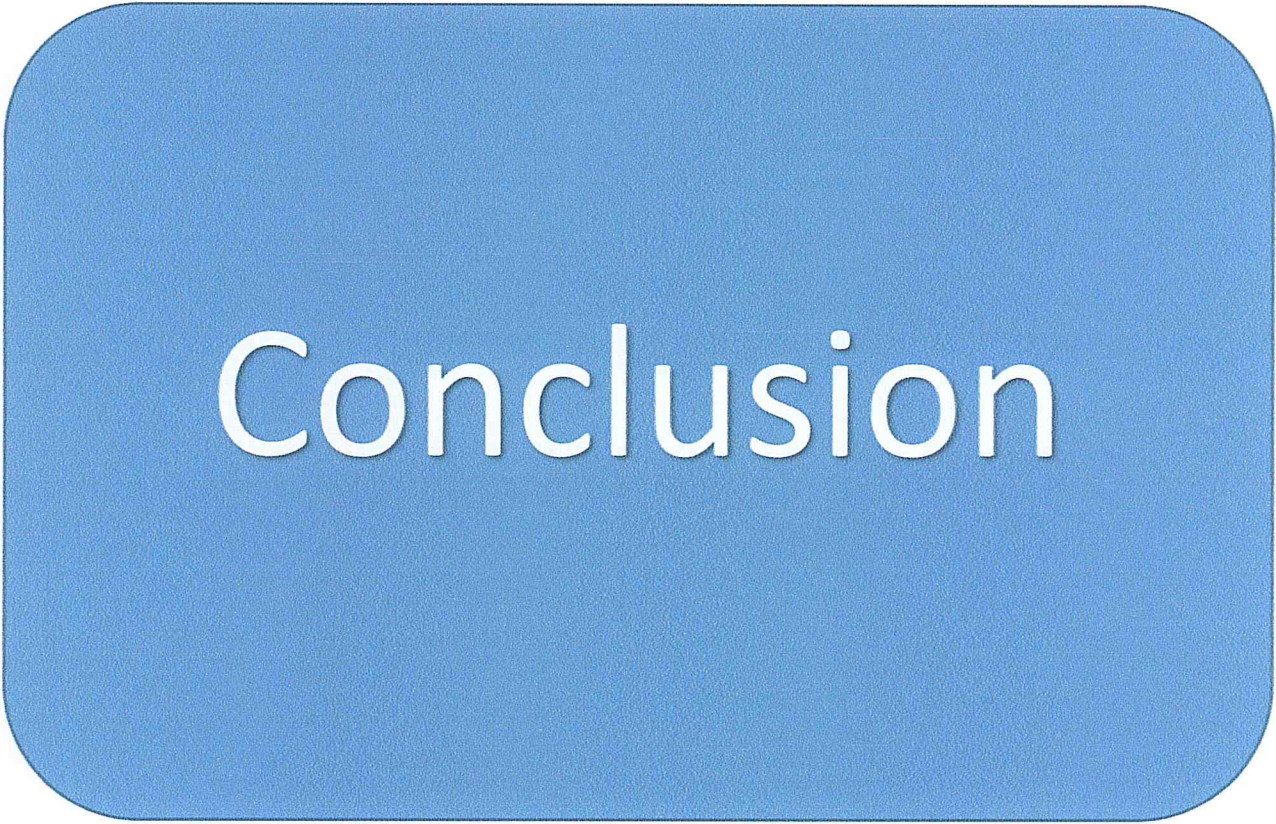
- Documentation incomplète sur le web : bien que Moodle soit l’une des plus fameuses plateformes d’enseignement, la documentation sur comment procéder est très pauvre, un développeur libre doit se casser la tête pour trouver les méthodes à appeler, les méthodes à redéfinir, etc. Moodle n’accepte un code PHP brute, certes on programmera avec PHP mais pas librement (en détails dans ce qui va suivre), Moodle contient des milliers ou plutôt des millions de lignes de codes, ainsi qu’une hiérarchie d’héritage de plusieurs dizaines de niveaux, que le développeur doit tout traquer pour pouvoir s’y trouver dans le code, une documentation simple et complète est vraiment indispensable pour mieux comprendre la logique de Moodle et savoir programmer à la Moodle.

- Interdiction d'utilisation des variables super-globales : PHP travaille avec des variables dites super-globales, tel que `$_POST`, `$_GET`, `$_SESSION`, etc. Moodle interdit le développeur de manipuler ses variables ou de les utiliser tout court limitant ainsi certaines utilisations, certes Moodle propose des méthodes et outils très utile mais ce point peut amener à revoir sa conception (s'il a été déjà faite).
- AJAX (Asynchronous JavaScript And XML) : est une architecture informatique puissante combinant à la fois le JavaScript, le DOM, JSON, XML, XMLHttpRequest, pour construire des applications web dynamique du coté client et rapporter des données d'une page en arrière-plan et sans y aller, pour utiliser AJAX sous Moodle il faut utiliser Core/AJAX un module de JavaScript (MoodleDocs), bien que cette bibliothèque en soit pas trop terrible, son utilisation requiert une bonne documentation que Moodle n'offre pas, Moodle spécifie seulement que core/AJAX est le seul qui peut être utilisé et comment l'installer et c'est tout!
- Les problèmes précédents nous ont poussés à utiliser des méthodes orthodoxes pour résoudre certains problèmes, par exemple comme on n'a pas pu utiliser le XMLHttpRequest pour envoyer en arrière-plan les données, on a choisi d'utiliser une méthode qui utilise un seul champ pour plusieurs données (synonyme par exemple, cf. page), ce qui en résulte une erreur lors de la modification d'une question, puisque pour modifier une question Moodle récupère toutes les données de tous les champs, mais vu que certains champs contiennent plusieurs données, donc forcément erreur.
- Pour finir développer sous Moodle requiert une concentration phénoménale, une simple ligne en plus, et rien ne s'affichera, il faut que tout soit parfait, ce qui est pas vraiment un inconvénient mais peut engendrer des situations désastreuses.

III.6. Conclusion :

La séparation MVC (module, vue, contrôle) nous était vraiment utile la séparation des algorithmes des interfaces peut faciliter la programmation et ainsi que la mise à jour du code.

Bien que l'intégration du module dans MOODLE n'était pas facile, certaines difficultés ont été rencontrées (voir page 79), mais elle s'est terminée avec succès ce qui prouve que MOODLE peut accepter de tels modules complexes.



Conclusion

V.1. Conclusion :

Le travail qui a été fait concerne l'évaluation automatique de questions courtes, un thème qui touche plusieurs domaines, le traitement de texte et de langues, la similarité, etc.

Nous avons commencé notre recherche par examiner et analyser les différentes approches et solutions utilisées pour résoudre le problème, qui se trouvent généralement dans des plateformes de télé-enseignement (e-learning), après l'analyse on a en tiré des avantages et des inconvénients de ces solutions, des aspects qui n'ont pas été pris en compte.

Afin d'éviter les problèmes rencontrés dans les solutions existantes, et surmonter les limites de ses solutions on a proposé une nouvelle approche basée sur la similarité de texte (dite distance) et quelques méthodes pour négliger l'aspect lexicale et syntaxique ne prendre en compte que l'aspect sémantique qui représente l'esprit de la réponse.

Ainsi toute chose qui n'appartient pas à la sémantique de la réponse doit être négligé, les espaces, les accents, les déterminants, etc.

Nous avons ensuite essayé d'intégrer ce module dans la plateforme MOODLE pour voir si la plateforme supportera un tel module complexe, les résultats étaient positives, le module a été intégré avec succès, ce qui peut nous donner une idée positive sur la plateforme MOODLE, et sa flexibilité en acceptant des tels modules.

Notre solution donne à l'enseignant la possibilité de créer des questions en engendrant un énorme nombre de possibilités, en déduisant ainsi les réponses, et les erreurs non-sémantiques (qui ne changeront pas le sens de la réponse) commises par l'étudiant, d'une manière souple et d'une facilité indéniable, en lui évitant toute notion qu'il ne peut maîtriser, l'outil a été développé de sorte que toute personne sache en servir.

V.2. Perspectives :

La recherche que nous avons faite a abouti à un résultat, mais l'outil reste loin de la correction humaine, l'IA s'avance lentement en essayant de simuler l'intelligence humaine, mais on est loin de cet objectif, pas pour les cinq prochaines années.

Dans ce qui va suivre, on évoquera des points qui n'ont pas été traités par l'outil :

- La négation : les deux phrases « Djamel est allé passer un séjour à Hassi bahbah » et « Djamel n'est pas allé passer un séjour à Hassi bahbah » sont proches en ce qui concerne la similarité de texte, mais différentes dans l'aspect sémantique.
- La polysémie : bien qu'on ait pu traiter les problèmes de synonymes et homonymes, la polysémie représente un vrai défi, quand un mot a plusieurs sens, il est très difficile de dire quel est le sens voulu selon le contexte.

Finalement on espère que notre solution a fait avancer la recherche dans le domaine de la correction automatique et qu'elle ouvre à d'autres développeurs le chemin vers des nouvelles idées et des nouvelles solutions.

Annexes

Annexe A :

Similarité NGram :

I. Code source (implémentation) :

```
public function getDistance( $source, $target) {
    $s1 = strlen($source);
    $t1 = strlen($target);
    if ($s1 == 0 || $t1 == 0) {
        if ($s1 == $t1) {
            return 1;
        }
        else {
            return 0;
        }
    }
    $cost = 0;
    if ($s1 < $this->n || $t1 < $this->n) {
        for ( $i=0, $ni= min($s1, $t1); $i<$ni; $i++) {
            if ($source{$i} == $target{$i}) {
                $cost++;
            }
        }
        return (float) $cost/max($s1, $t1);
    }
    $sa = array();
    $p = array();
    $d = array();
    $_d = array();
    for ( $i=0; $i<($s1+$this->n-1); $i++ ){
        if ($i < $this->n-1) {
            $sa[$i]=0;
        }
        else {
            $sa[$i] = $source{$i-$this->n+1};
        }
    }
}
```

```
$p = array();
$d = array();
si;
  sj;

  $t_j = array();

for (si = 0; si<=$s1; si++) {
  $p[si] = si;
}
for (sj = 1; sj<=$t1; sj++) {

  if (sj < $this->n) {
    for ( sti=0; sti<$this->n-sj; sti++) {
      $t_j[sti]=0;
    }
    for ( sti=$this->n-sj; sti<$this->n; sti++) {
      $t_j[sti]=$target{sti-($this->n-sj)};
    }
  }
  else {

    $t_j = str_split(substr($target, sj-$this->n, $this->n));
  }
  $d[0] = sj;
  for (si=1; si<=$s1; si++) {
    $cost = 0;
    $tn=$this->n;
```



```

    for ( $ni=0;$ni<$this->n;$ni++) {

        if ($sa[$i-1+$ni] != $t_j[$ni]) {
            $cost++;
        }
        else if ($sa[$i-1+$ni] == 0) {
            $tn--;
        }
    }
    $ec = (float) $cost/$tn;

    $d[$i] = min(min($d[$i-1]+1, $p[$i]+1), $p[$i-1]+$ec);
}

$_d = $p;
$p = $d;
$d = $_d;
}

return 1.0 - ($p[$s1] / max($t1, $s1));
}

```

II. Exemple :

Un exemple pour les trous algorithm sera mis à la fin.

Annexe B :

Similarité Levenstein :

I. Code source (implémentation) :

```
public function getDistance ($target, $other) {
    $sa = str_split($target);
    $n = count($sa);
    $p = array();
    $d = array();
    $m = strlen($other);
    if ($n == 0 || $m == 0) {
        if ($n == $m) {
            return 1;
        }
        else {
            return 0;
        }
    }
    for ($i = 0; $i <= $n; $i++) {
        $p[$i] = $i;
    }

    for ($j = 1; $j <= $m; $j++) {
        $t_j = $other[$j-1];
        $d[0] = $j;

        for ($i=1; $i <= $n; $i++) {
            $cost = $sa[$i-1] == $t_j ? 0 : 1;
            $d[$i] = min(min($d[$i-1]+1, $p[$i]+1), $p[$i-1]+$cost);
        }
        $_d = $p;
        $p = $d;
        $d = $_d;
    }

    return 1.0 - ((float) $p[$n] / max(strlen($other), count($sa)));
}
```

II. Exemple :

Un exemple pour les trous algorithmes sera mis à la fin.

Annexe C :

Similarité JaroWinkler :

I. Code source (implémentation) :

```
function matches($s1,$s2) {
    $max="";
    $min="";
    if (strlen($s1) > strlen($s2)) {
        $max = $s1; $min = $s2; } else {
        $max = $s2; $min = $s1; }
    $range = max((int)(strlen($max) / 2) - 1, 0);
    $matchIndexes = array_fill(0,strlen($min),-1);
    $matchFlags = array_fill(0,strlen($max),0);
    $matches = 0;
    for ( $mi = 0; $mi < strlen($min); $mi++) {
        $c1 = $min[$mi];
        for ( $xi = max($mi - $range, 0), $xn = min($mi + $range + 1, strlen($max)); $xi < $xn; $xi++) {
            if (!$matchFlags[$xi] && $c1 == $max[$xi]) {
                $matchIndexes[$mi] = $xi;
                $matchFlags[$xi] = true;
                $matches++;
                break;}}
    }
    $ms1=array(); $ms2=array();
    for ( $i = 0, $si = 0; $i < strlen($min); $i++) {
        if ($matchIndexes[$i] != -1) {
            $ms1[$si] = $min($i);
            $si++; } }
    for ( $i = 0, $si = 0; $i < strlen($max); $i++) {
        if ($matchFlags[$i]) {
            $ms2[$si] = $max($i);
            $si++; } }
    $transpositions = 0;
    for ( $mi = 0; $mi < count($ms1); $mi++) {
        if ($ms1[$mi] != $ms2[$mi]) {
            $transpositions++; } }
    $prefix = 0;
    for ( $mi = 0; $mi < strlen($min); $mi++) {
        if ($s1[$mi] == $s2[$mi]) {
            $prefix++; } else {
            break; } }
    return array ( $matches, (int)($transpositions / 2), $prefix, strlen($max) );
}

function getDistance( $s1, $s2) {
    $mtp = $this->matches($s1, $s2);
    $m = $mtp[0];
    if ($m == 0) {
        return 0;
    }
    $j = (($m / strlen($s1) + $m / strlen($s2) + ($m - $mtp[1]) / $m) / 3;
    $jw = $j < $this->getThreshold() ? $j : $j + min(0.1, 1 / $mtp[3]) * $mtp[2] * (1 - $j);
    return $jw;
}
```

II. Exemple :

Un exemple pour les trous algorithmme sera mis à la fin.

Annexe D :

Exemple :

1. Calculons le taux de similarité entre « espaces pris en compte » et « espaces pris en compte » :

NGram : 0.32
Levenstein : 0.34
JaroWinkler : 0.64952240802676



Comme on peut voir dans les résultats, les espaces sont pris en compte, donc il faut bien les éliminer pour avoir un résultat efficace.

2. Calculons le taux de similarité entre « cet été était génial » et « cet ete etait genial » :

NGram : 0.76190476190476
Levenstein : 0.76190476190476
JaroWinkler : 0.88178382464097

Même chose sera site pour les accents.

3. Calculons le taux de similarité entre « le chat est dans la maison » et « un chat es dans le maison » :

NGram : 0.81481481481481
Levenstein : 0.85185185185185
JaroWinkler : 0.81232193732194

4. Calculons le taux de similarité entre « la casse est importante » et « LA CASSE EST IMPORTANTE » :

NGram : 0.1304347826087
Levenstein : 0.1304347826087
JaroWinkler : 0.42028985507246

Conclusion :

Bien que ces algorithmes de similarité soient différents mais ils prennent en compte tous les changements possibles, espaces, la casse, les déterminants, donc on a bien fait de les négliger dans notre outil.