HAL open science

Intégration de raisonnements automatiques et de contenu informatif pour la synthèse d'erreurs récurrentes dans des compte-rendus de TP

Hakim Mokeddem, Cyrille Desmoulins

▶ **To cite this version:**

Hakim Mokeddem, Cyrille Desmoulins. Intégration de raisonnements automatiques et de contenu informatif pour la synthèse d'erreurs récurrentes dans des compte-rendus de TP. Journée EIAH&IA 2013, May 2013, Toulouse, France. 12 p. hal-00824293

HAL Id: hal-00824293

https://hal.science/hal-00824293v1

Submitted on 21 May 2013

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

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

Intégration de raisonnements automatiques et de contenu informatif pour la synthèse d'erreurs récurrentes dans des compte-rendus de TP

Hakim Mokeddem[1], Cyrille Desmoulins[2]

[1] Ecole Supérieure d'Informatique ESI, Alger
h_mokeddem@esi.dz
[2] Laboratoire d'Informatique de Grenoble, Université Joseph Fourier,
cyrille.desmoulins@imag.fr

Résumé. Cet article traite de l'intégration de MemoNote, un outil d'annotation basé sur les technologies du web sémantique, dans LabBook, un système de rédaction de compte- rendus de TP, pour offrir aux utilisateurs des services d'annotations sémantique. Il présente en premier lieu l'étude menée auprès d'enseignants utilisateurs de LabBook pour définir ces nouveaux services. La fonctionnalité la plus intéressante pour les enseignants, le calcul du type d'erreur le plus courant, est ensuite définie avec précision à la fois comme le type d'erreur le plus spécifique et couvrant le plus grand nombre d'instances. Elle nécessite de prendre en compte la structure hiérarchique des différents types d'erreurs dans une ontologie. Pour cela, nous avons proposé un algorithme basé à la fois sur un raisonnement automatique dans l'ontologie OWL DL des objectifs, basé sur la "réalisation" automatique des instances, et sur le contenu informatif (utilisé en recherche d'information). L'application de l'algorithme sur un exemple montre que les règles définies pour répondre à ce besoin sont vérifiées.

Mots-clés: Annotation, ontologie, travaux pratiques, raisonnement automatique, contenu informatif.

1 Introduction

Notre équipe de recherche a développé indépendamment deux systèmes différents. D'un côté, le système d'apprentissage sur le WEB LabBook permet aux étudiants en sciences expérimentales de créer des compte-rendus de travaux pratiques et de les envoyer pour évaluation aux enseignants. Les évaluations sur ces documents comprennent des commentaires textuels, mémorisant les retours que souhaite donner l'enseignant à l'étudiant sur un passage de son compte-rendu, par exemple lui signaler une erreur ou un manque. D'un autre côté, l'outil d'annotation MemoNote [1] permet à des enseignants de créer des annotations sur un document du web en leur donnant une sémantique formelle. Basé sur les technologies du web sémantique [3] (OWL [5] et les outils de raisonnement et de requête associés), il prend la forme d'une extension

du navigateur web Mozilla Firefox. Une ontologie des annotations selon objectifs donne du sens aux annotations, en référence aux types d'erreurs de l'apprenant.

Intégrer MemoNote dans LabBook [4] permet d'accéder à la sémantique des annotations, ce qui est difficile avec des annotations textuelles, et permet d'offrir de nouveaux services spécifiques aux enseignants, ce que ne fournissent pas les outils d'annotation, sémantiques ou non [2]. Pour établir les services utiles aux enseignants à partir de cette sémantique, nous avons mis en placfe une démarche expérimentale de capture des besoins d'enseignants de chimie utilisant LabBook pour des TPs à l'université. Cet article développe comment l'utilisation conjointe de techniques du web sémantique (via des raisonneurs sur les ontologies) et de techniques de recherche d'information (basées sur le contenu informatif), permet de répondre à ces besoins.

L'article est organisé de la façon suivante. La deuxième section décrit la démarche expérimentale d'extraction des besoins. La troisième section présente les besoins issus de cette étude. La quatrième développe le problème posé par un besoin particulier (calcul de l'erreur la plus courante). La cinquième section propose une solution pour ce calcul et montre qu'elle correspond bien au besoin décrit.

2 Démarche expérimentale pour définir les besoins

La démarche expérimentale de définition des besoins a trois principaux objectifs. Le premier est de valider auprès des enseignants les fonctionnalités de base de l'outil MemoNote intégrées à LabBook, le deuxième est de vérifier l'utilité de nouvelles fonctionnalités proposées spécifiquement pour LabBook, enfin le troisième objectif est de recueillir de nouveaux besoins dans le cadre de LabBook.

Pour mener cette étude, nous avons défini un protocole basé sur des méthodes qualitatives. L'étude est basée sur des entretiens individuels avec des enseignants de chimie ayant déjà une expérience dans LabBook et d'annotation papier. Chaque entretien est guidé par des maquettes d'interfaces des fonctionnalités possibles. Le but de ces maquettes est de mettre l'enseignant dans une situation réelle d'utilisation et de le faire réagir sur cette situation et indirectement sur les fonctionnalités de l'outil.

2.1 Procédure globale

L'expérience est faite avec 4 enseignants de l'université de Grenoble utilisant LabBook dans un enseignement en sciences expérimentale de 1ère année de licence.

Le protocole expérimental est divisé en quatre étapes. La première étape consiste à présenter à chaque enseignant l'objectif de l'expérience. La deuxième étape consiste à parcourir avec chaque enseignant les maquettes graphiques des fonctionnalités de base de MemoNote. Enfin, dans la quatrième étape les enseignants décrivent leurs autres besoins d'annotation dans LabBook sur la base des fonctionnalités proposées.

A la fin de chaque étape, un ensemble de questions est posée aux enseignants sur l'utilité de l'outil, les fonctionnalités et des suggestions pour améliorer l'outil et offrir des nouvelles fonctionnalités.

2.2 Les fonctionnalités de base présentées

MemoNote permet aux utilisateurs d'annoter des documents textuels, les trois fonctionnalités principales de MemoNote présentées sont les suivantes :

1. Ouvrir une séance. Après l'authentification le système demande à l'enseignant de spécifier les différents éléments de la séance en cours comme le lieu, le domaine d'apprentissage, le niveau d'apprentissage et l'activité d'apprentissage .
2. Annoter sans patron. L'enseignant sélectionne une forme graphique, un contenu à annoter et spécifie ensuite l'objectif de l'annotation et ses éléments en naviguant dans la hiérarchie des objectifs et les éléments d'objectif (tous deux représentés dans une ontologie), puis la force et l'importance de l'annotation.
3. Annoter avec patron. L'enseignant sélectionne une forme graphique, un contenu à annoter. Le système détecte un patron d'annotation correspondant à cette forme graphique dans ce contexte (défini par les éléments de la séance) et en déduit objectif, force et importance de l'annotation suivant le patron. Il reste à l'enseignant à donner le sous-type précis d'objectif et à indiquer un élément d'objectif. Par exemple, le patron signaler une erreur correspond à la forme graphique surligner en rouge dans la correction d'un compte-rendus de TP de chimie. Dans ce contexte, lorsque l'utilisateur sélectionne cette forme graphique, le système lui propose de choisir un sous-objectif de l'objectif "signaler une erreur" (par exemple "Signaler une erreur algébrique" et de designer comme élément d'objectif "exponentielle sur logartithme" et génére le reste de l'annotation automatiquement.

MemoNote mémorise les annotations à partir d'une ontologie des annotations. Elle permet de représenter les dimensions suivantes de l'annotation de l'enseignant :

1. La séance d'annotation : lieu, date début et de fin, auteur, activité d'apprentissage et activité d'enseignement, domaine et niveau d'apprentissage.
2. La forme d'annotation : forme graphique, couleur et ancre.
3. L'objectif d'annotation qui donne son sens à l'annotation, sa force et son importance.

Chaque annotation est une instance de l'une des trois sous-classes d'annotations :

1. Annotation pédagogique représente l'avis pédagogique de l'enseignant sur l'activité d'apprentissage.
2. Annotation domaine représente le point de vue de l'enseignant sur le document dans un domaine d'apprentissage.
3. Annotation document représente l'avis de l'enseignant sur la façon de décrire l'activité pédagogique et domaine sur le document.

Chacun annotation est reliée, suivant son type, à un élément d'objectif pris dans une ontologie dédiée. Par exemple une annotation qui sert à mentionner une erreur est une instance de la classe *mention_An_Error* qui elle même une sous classe de annotation domaine, la classe *mention_An_Error* possède des éléments d'objectifs représentés dans une hiérarchie des erreurs possibles.

2.3 Les fonctionnalités nouvelles pour LabBook

Des scénarios de deux nouvelles fonctionnalités de MemoNote sous LabBook sont présentés aux enseignants :

1. Calcul du type d'erreur la plus courante, qui consiste à afficher à l'enseignant le type d'erreur le plus annoté par l'ensemble des enseignants dans les compte-rendus d'un TP particulier.
2. Partage des annotations, qui permet à un enseignant de partager des annotations avec ses collègues.

3 Résultats de l'étude

L'étude a permis de dégager en premier lieu, les avis des enseignants sur l'outil MemoNote et son intégration dans LabBook, puis les retours des enseignants sur les nouvelles fonctionnalités proposées et enfin les nouveaux besoins en services d'annotations exprimés par les enseignants.

3.1 Fonctionnalités de base intégrées à LabBook

Les retours des enseignants sur l'intégration de MémoNote à LabBook sont les suivants :

1. L'interface homme machine est souvent critiquée par les utilisateurs, notamment la visualisation des informations de chaque annotation et l'utilisation des crayons de la barre d'outils pour annoter.
2. L'ontologie des objectifs d'annotation est à affiner dans ce contexte de corrections de compte-rendus de TP. Certains éléments ne correspondent pas à des cas réels dans LabBook, par exemple le concept mentionner une erreur de manipulation n'est pas utilisé souvent, car l'enseignant ne peut pas savoir en corrigeant un compte-rendu que l'étudiant a fait une erreur de manipulation.

Globalement l'étude a montré l'intérêt pour les enseignants d'utiliser les annotations de MemoNote dans LabBook surtout avec l'utilisation facile de l'outil pour créer des annotations et son intégration rapide avec LabBook en installant une extension du navigateur Firefox.

3.2 Fonctionnalités nouvelles pour LabBook

Le "calcul du type d'erreurs le plus fréquent" intéresse le plus les enseignants, car elle leur permet de donner des retours qualitatifs aux étudiants sur l'évaluation de leur TPs, de se faire une idée sur les problèmes rencontrés par les étudiants et d'établir un bilan général sur l'enseignement de la matière. L'autre nouvelle fonctionnalité ("partage des annotations") n'a pas été beaucoup appréciée par les enseignants, car les annotations sont partagées a priori sur LabBook par tous les enseignants.

3.3 Nouveaux besoins pour LabBook

Les nouvelles fonctionnalités proposées par les enseignants sont la création automatique des patrons et la reproduction des annotations récentes, en vue de leur faciliter la tâche d'annotation. La première offre une création automatique de patron en fonction des annotations précédentes de l'enseignant. La seconde reproduit la dernière annotation et laisse l'enseignant la modifier si besoin. D'autres fonctionnalités intéressent les enseignants comme la liaison les annotations et une grille d'évaluation contenant des compétences à acquérir lors d'un TP, la recherche sur une étape particulière du TP et l'adaptation des patrons en fonction du TP, par exemple signaler des erreurs particulières pour un TP.

Nous détaillons maintenant le besoin le plus intéressant pour les enseignants : le calcul du type d'erreurs le plus courant. Nous commençons par présenter le problème de ce calcul pour ensuite présenter notre solution à ce problème.

4 Problème de calcul du type d'erreur le plus courant

Pour définir ce qu'est le type d'erreur le plus courant dans ce contexte, il faut prendre en compte à la fois les erreurs signalées et leur niveau de généralité. Ce n'est pas l'erreur la plus courante par référence à une liste d'erreurs possibles, comme ce serait possible avec des erreurs répertoriées dans une base de données, mais en référence à la taxinomie des types d'erreurs et aux types d'erreur effectivement faites par les étudiants. En particulier, l'enseignant peut signaler des erreurs plus ou moins générales à l'apprenant. Le type d'erreur le plus courant peut être très spécifique (en bas de taxinomie des types d'erreurs) ou assez général (plus haut dans la taxinomie des types d'erreurs) selon la répartition des erreurs signalées effectivement aux apprenants par les enseignants.

Le type d'erreur le plus courant est alors défini par la classe d'erreur la plus focalisée possible contenant le plus d'erreurs signalées comparativement aux autres classes d'erreurs.

La classe d'erreur la plus générale correspond rarement à cette définition, car c'est la moins focalisée. Ce cas indique qu'il n'y a pas de type d'erreur le plus courant, car tous les types d'erreurs sont annotés de façon répartie. Les classes d'erreurs les plus spécifiques (en "bas" de la taxinomie) correspondent aussi rarement, car les erreurs dans classes d'erreurs voisines viennent contribuer à la classe d'erreur juste supérieure. Le plus souvent, il s'agit donc de trouver là où les classes d'erreurs les plus courantes, entre ces deux extrêmes. L'exemple ci-dessous montre un exemple de la hiérarchie des erreurs annotées.

Les différents types d'erreur qui sont des éléments d'objectif de la classe *mention_An_Error* sont représentés dans la hiérarchie décrite dans la figure 1.

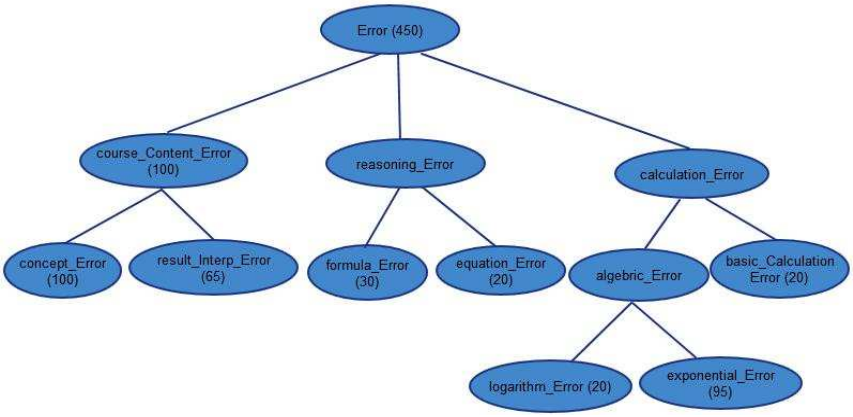

Fig. 1. Ontologie des erreurs avec des instances

La première manière de faire est d'utiliser une généralisation simple en calculant la hiérarchie des classes et des instances et ensuite prendre la classe qui possède le nombre d'instance le plus élevé, mais le problème avec cette méthode c'est que le type d'erreur le plus fréquent est toujours la racine de l'arbre.

La deuxième manière utilise le contenu informatif [7] de chaque nœud sans prendre en considération le nombre d'instance, l'idée du contenu informatif est la suivante : plus une classe est haute dans l'arbre des sous-classes moins elle est informative (plus abstrait). Le contenu informatif est ainsi est une fonction croissante de la racine vers les feuilles. Avec cette méthode, les feuilles de l'arbre les plus profondes sont favorisées sans tenir compte des instances, les autres niveaux de l'arbre sont ignorés.

Aucune des deux méthodes ne correspond à ce qui est attendu ; la première donne un résultat détaillé, mais ignore la relation sémantique de sous-classe entre les classes, la deuxième donne un résultat très abstrait.

Notre objectif est donc de trouver une solution entre les deux précédentes, c'est-à-dire de calculer les erreurs les plus courantes en tenant compte des relations sémantiques entre les classes.

5 Un calcul basé sur des raisonnements et le contenu informatif

Pour prendre considération à la fois les erreurs signalées et leur niveau de généralité, nous proposons une solution basée sur la "réalisation" des instances de l'ontologie (pour les erreurs signalées) et du contenu informatif des concepts (pour le niveau de généralité). Nous décrivons maintenant l'intérêt d'utiliser ces deux solutions.

5.1 La "réalisation" des instances

La "réalisation" automatique des instances calculer tous les types possibles de chaque instance, et par conséquent le nombre d'instances de chaque concept. Par exemple, la "réalisation" automatique des instances pour l'exemple de la section précédente,

donne les nombres d'instances pour chaque concept données dans la figure 2. Le concept Course_Content_Error a été annoté 100 fois initialement. Après "réalisation" le concept comprend aussi les instances de ses sous classes (*concept_Error* et *result_Interp_Error*).

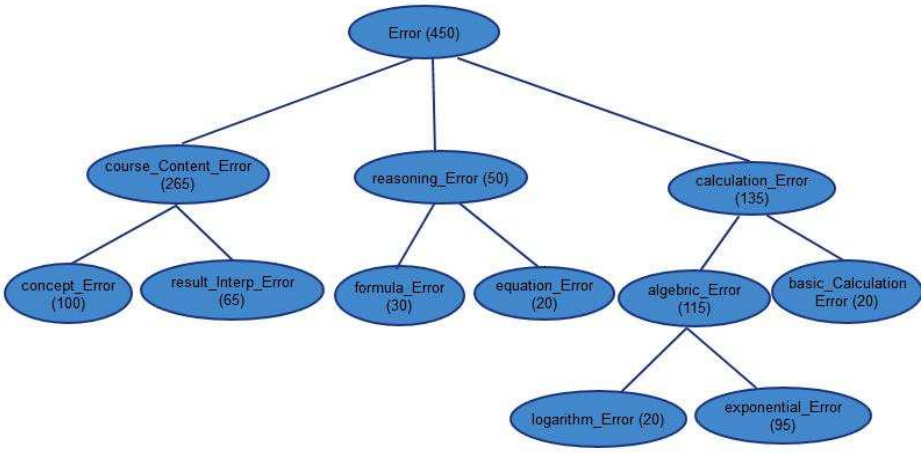

Fig. 2. Ontologie des types d'erreurs classifiée

5.2 Calcul du contenu informatif d'un concept (Information Content IC)

Nous précisons ici le calcul du contenu informatif d'un concept dans une hiérarchie des concepts.

Le contenu informatif est introduit pour mesurer la similarité entre deux concepts dans une taxonomie WordNet [6]. La similarité est utilisée en recherche d'information : si deux concepts sont sémantiquement similaires alors ils sont classés de la même façon par les recherches.

La similarité dans une taxonomie WordNet est alors le contenu informatif du plus petit concept commun (qui généralise les deux concepts). Le contenu informatif d'un concept est calculé en utilisant la probabilité d'apparition de l'ensemble de ses sous concepts dans le corpus d'apprentissage, soit c = -Log(p(c)) où p(c) est la probabilité de trouver une instance d'un concept c dans le corpus. Plus on monte dans la hiérarchie plus on a de chance de rencontrer le concept dans le corpus et donc le contenu informatif diminue à cause du log entre 0 et 1.

Plus le concept est précis, plus son contenu informatif est grand. Pour le calculer, on utilise la probabilité d'appartenance à un corpus d'apprentissage c = -Log(p(c)) où p(c) est la probabilité de trouver une instance d'un concept c dans le corpus. Plus on monte dans la hiérarchie plus on a de chance de rencontrer le concept dans le corpus et donc le contenu informatif diminue.

Il existe une autre formule proposée par [8] pour calculer le contenu informatif d'un concept en tenant compte des nombres du nombre des descendants. L'idée de

cette formule est la suivante : plus un concept a des descendants moins il est informatif, cette formule utilise donc les hyponymes pour calculer le degré informatif :

$$IC(c) = \frac{\log(\frac{hypo(c)+1}{\max(w_n)})}{\log(\frac{1}{\max(w_n)})} = (1 - \frac{\log(\mathrm{hypo(c)+1})}{\log(\max(w_n))}) \quad (1)$$

hypo(c) indique le nombre d'hyponymes (descendant) dont possède l'individu C
max(Wn) est le nombre maximum de concept dans l'ontologie.

La formule générale du IC proposé par [9] prend en compte la profondeur de l'arbre et le nombre de descendants, en effet plus le concept est profond et possède moins de descendants alors plus il est informatif. La formule générale du IC est la suivante :

$$IC(c) = \mathrm{k}\,(1 - \frac{\log(\mathrm{hypo(c)+1})}{\log(\max(\mathrm{wn}))}) + (1-k)(\frac{\log(\mathrm{deep(c)})}{\log(\mathrm{deep_{max}})}) \quad (2)$$

deep(c) est la profondeur du concept c dans l'arbre est la profondeur maximale de l'arbre.

K est un paramètre pour ajuster le poids des deux éléments de l'équation. Le premier élément de l'équation favorise les classes ayant moins de fils et le deuxième élément de l'équation favorise les classes profondes dans la hiérarchie.

- k = 1 favorise les concepts ayant les moins de descendants sans tenir compte sa profondeur dans l'arbre.
- k = 0 favorise la profondeur de l'élément dans l'arbre sans tenir pas compte du nombre de ses fils
- k = 0.5 favorise à part égales ces deux éléments.

5.3 Calcul de l'erreur la plus courante

Nous définissions maintenant des règles pour calculer le type d'erreur le plus courante dans l'ontologie des erreurs

Règles pour le calcul du type d'erreur le plus courant

La définition du type d'erreur le plus courant peut être reformulée par les deux règles suivantes :

Nombre important d'erreurs. Plus un type d'erreur contient d'instances plus il est courant.

Contenu informatif élevé. Plus le contenu informatif d'un type d'erreur est élevé, plus le type d'erreur est courant

L'objectif est donc de définir un calcul du type d'erreur qui réponde à la fois à ces deux règles, c'est-à-dire ayant à la fois un nombre important d'erreurs et un contenu informatif élevé.

Il faut pour cela définir une formule de calcul qui combine le calcul du contenu informationnel et le nombre d'instance et vérifier qu'elle correspond bien aux deux règles précédentes.

Évaluation de la règle *nombre important d'erreurs*

Pour évaluer le nombre important d'erreurs, on utilise la fréquence de chaque concept. La fréquence d'un type d'erreur C est égale au rapport entre le nombre d'instances de C et le nombre global d'instances dans l'ontologie.

Soit C un type d'erreur donnée.

$$\mathrm{Freq(C)} = \frac{\text{Nombre d'instance de C}}{\text{Nombre global d'instance}} \quad (3)$$

Évaluation de la règle Contenu informatif élevé

Nous découpons la règle *Contenu informatif élevé* en deux règles auxquelles il est possible de rattacher une formule de calcul, qui sont les suivantes :

Profondeur importante. Plus un concept est profond dans l'arbre plus il est informatif. En effet, plus le type d'erreur est profond plus il représente une erreur précise. Par exemple, dans l'ontologie de la figure 2, le type d'erreur "application du logarithme" représente une erreur détaillée, par contre "erreur algébrique" est plus générale et peut représenter plusieurs types spécifiques.

Faibles détails. Moins un concept a de sous-classes (sous-classes directes et leurs sous classes) plus il est informatif.

Ainsi, un concept ayant plusieurs sous classes, perd de l'information dans les sous classes. Par exemple, le type d'erreur "erreur de calcul " perd de l'information au profit de ces quatre descendants, c'est-à-dire que les enseignants cherchent toujours à spécifier une erreur plus détaillée d'un type donné.

Soit C un type d'erreur donnée.

La formule de IC(C) = formule précédente avec k = 0,5 qui représente à la fois les deux règles, compromis entre profondeur importante et faible détails.

Formule de calcul de l'erreur la plus courante

À partir des deux formules précédentes, nous proposons la formule ci-dessous qui permet de calculer la récurrence d'un type d'erreur le plus courant en tenant de compte des deux règles (nombre important d'erreurs et contenu informatif élevé) à la fois.

Récurrence(c) = min (IC(c), Freq(c))

où IC(c) est le contenu informatif du concept c et Freq(c) est la fréquence du concept c dans la hiérarchie.

La formule de calcul est un minimum pour favoriser à la fois le contenu informatif et la fréquence. Prendre le minimum permet de mettre en avant les types d'erreur qui ont à la fois un contenu informatif important et une fréquence importante. En effet, si le résultat est fort, c'est que les deux valeurs étaient fortes, car dès qu'une des deux

valeurs est faible, le résultat est faible. Le type d'erreur le plus courant est le type pour lequel la fonction *Récurrence* est maximale.

5.4 Évaluation

Nous appliquons l'algorithme de calcul du type d'erreur le plus courant sur l'exemple présenté précédemment (section 4 et section 5.1). On applique la formule de *Récurrence* sur l'exemple de la figure 2 et obtient la récurrence de chaque type d'erreur comme indiqué dans la figure 3. (*Réc* est la récurrence d'un concept)

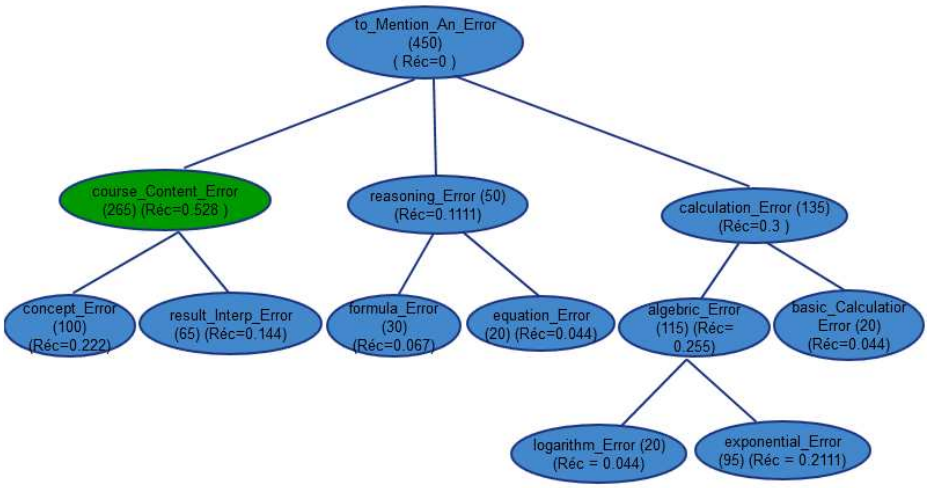

Fig. 3. Type d'erreur le pus fréquent

A partir des règles définies, le résultat obtenu est conforme aux attentes. Le type d'erreur le plus courant est la classe la plus annotée et la plus focalisée possible par rapport aux classes voisines, la classe *course_Content_Error* est la classe qui possède le plus grand nombre d'instances avec un contenu informationnel important aussi.

La formule utilisée (maximum des minima entre contenu informatique et fréquence d'annotation) est une formule couramment utilisée en recherche d'information.

L'algorithme proposé prend en compte trois paramètres pour calculer le contenu informatif d'un concept donné : le nombre d'instances, le rapport entre sa profondeur et la profondeur maximale de l'arbre et le rapport entre le nombre de ses sous concepts et le nombre total des concepts. La hauteur de l'arbre est un autre facteur:

- Plus la profondeur de l'arbre est grande, plus les concepts les plus spécifiques sont favorisés, car le contenu informatif des classes les plus générales devient faible.
- Plus la profondeur de l'arbre est petite, plus les concepts les plus génériques ayant moins de sous concepts sont favorisés, car leur contenu informatif reste important.

L'algorithme proposé fonctionne bien dans le cas ou l'arbre n'est pas très profond, car il n'élimine pas à priori aucun niveau dans la hiérarchie. Dans le cas ou l'arbre est

très profond, un ajustement de la formule doit se faire afin de ne pas pénaliser les concepts génériques qui ont un nombre d'instances important.

6 Conclusion et perspectives

L'intégration de l'outil d'annotation MemoNote avec le système de rédaction de compte-rendus en science expérimentale LabBook permet d'offrir de nouveaux services aux enseignants, car les technologies du web sémantique sur lesquelles MemoNote est basé permet d'accéder à la sémantique des annotations.

Pour identifier les services d'annotations utiles pour les enseignants avec l'utilisation de LabBook, nous avons élaboré une étude auprès des enseignants utilisant LabBook pour la correction de TPs en science expérimentale. Les résultats de l'étude montrent que le cas d'utilisation qui intéresse plus les enseignants est le calcul du type d'erreur le plus fréquent, qui permet aux enseignants de faire un bilan général sur les problèmes rencontrés par les étudiants dans un TP particulier. Le calcul du type d'erreur le plus fréquent est complexe et ne revient pas à trouver l'erreur la plus annotée comme dans une base de données, mais nécessite de prendre en compte la structure ontologique des différents types d'erreurs et consiste à trouver la classe la plus spécifique dans la hiérarchie ayant un grand nombre d'instances.

Pour calculer le type d'erreur le plus courant, nous avons proposé un algorithme basé sur le raisonnement automatique OWL DL et le contenu informatif utilisé en recherche d'information.

Le raisonnement automatique est basé sur la "réalisation" automatique des instances qui sert à calculer les différents types de chaque instance afin de trouver le nombre total d'instances pour chaque classe. La fréquence de chaque concept est ensuite calculée pour l'utiliser dans la formule globale de calcul du type d'erreur le plus courant.

Le contenu informatif est une valeur utilisée en recherche d'information pour mesurer le degré informationnel des concepts dans une hiérarchie. Cette mesure augmente dans les concepts les plus spécifiques et diminue dans les classes les plus générales. Pour répondre à notre cas d'utilisation, nous avons choisi une formule pour calculer le contenu informatif qui prend en compte à la fois la profondeur de chaque concept dans l'arbre et le nombre de ses sous concepts. Ainsi, plus un concept est profond plus il est informatif, moins il a des sous concepts plus il est informatif.

Le type d'erreur le plus courant est calculé par la fonction de récurrence de chaque type en prenant le minimum entre sa fréquence et le contenu informatif. Le type d'erreur le plus fréquent est le concept ayant le maximum d'occurrence.

L'évaluation de l'algorithme est basée sur la vérification des règles définies. Le test effectué de l'algorithme sur un exemple montre que le résultat obtenu est satisfaisant. Le type d'erreur le plus fréquent est le concept ayant le maximum de récurrence correspond aux formules couramment utilisées en recherche d'information avec des sous-types de concepts.

Les questions qui restent à développer concernent principalement l'intégration de cette fonctionnalité dans l'extension de Firefox afin de permettre aux utilisateurs de la

tester. Il sera alors possible de comparer les types d'erreurs produits par notre algorithme avec celui attendu par les enseignants.

L'autre point qui reste à développer est de développer les besoins les plus importants définis par les enseignants. Par exemple, la création automatique des patrons qui pourrait utiliser des techniques similaires au type d'erreur le plus fréquent, car il s'agira alors du type d'annotation le plus fréquent d'un enseignant. La difficulté supplémentaire est de généraliser plusieurs arbres de sous-concepts à la fois, en particulier pour déterminer le contexte du patron qui généralise l'ensemble des séances, tout en restant suffisamment focalisé.

7 Références

1. Azouaou, F., Desmoulins, C.: MemoNote, A context-aware annotation tool for teachers. 7th International Conference on Information Technology Based Higher Education and Training, 2006. ITHET'06. pp. 621–628. , Sydney, Australia (2006).
2. Azouaou, F., Weiqin, C., Desmoulins, C.: Semantic Annotation Tools for Learning Material. Proceedings of Semantic Web and e-learning Workshop, Vol. 2, CS-Report 04-19, Adaptive Hypermedia (AH'04), Eindhoven University of Technology. pp. 359–364 (2004).
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific american. 284, 28–37 (2001).
4. d' Ham, C., Girault, I., Bodin, M.: Des outils de support à l'investigation scientifique : intégration dans les plates-formes SCY-Lab et LabBook. Presented at the EIAH 2011 : Environnements Informatiques d'Apprentissage Humain , Mons, Belgique 27 mai (2011).
5. McGuinness, D.L., Van Harmelen, F.: OWL web ontology language overview. W3C recommendation. 10, 10 (2004).
6. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM. 38, 39–41 (1995).
7. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In Proceedings of the 14th International Joint Conference on Artificial Intelligence. pp. 448–453 (1995).
8. Seco, N., Veale, T., Hayes, J.: An Intrinsic Information Content Metric for Semantic Similarity in WordNet. Presented at the ECAI'2004, the 16th European Conference on Artificial Intelligence , Amsterdam Netherlands (2004).
9. Zhou, Z., Wang, Y., Gu, J.: A New Model of Information Content for Semantic Similarity in WordNet. Second International Conference on Future Generation Communication and Networking Symposia, 2008. FGCNS'08. pp. 85 –89 (2008).