

Analyse des traces de LIF3

Contexte d'usage et scénarios d'analyse

[Cas d'usage LIF3](#)

[Contexte](#)

[Trace](#)

[Scénarios d'analyse LIF3](#)

[Sc1 : qualité des QCM](#)

[Sc1a : Taux de réussite pour une question de QCM](#)

[Sc1b : Taux d'utilisation d'un couple question-distracteurs](#)

[Sc2 : Identifier les "types" d'apprenants](#)

[Sc2a : Quand les apprenants utilisent-ils la plateforme ?](#)

[Sc2b : Comment les apprenants utilisent-ils la plateforme \(plusieurs essais sur la même version du même exercice\) ?](#)

[Sc2c : Comment les apprenants utilisent-ils la plateforme \(plusieurs versions du même exercice\) ?](#)

[Sc2d : Pourquoi du comment les apprenants utilisent la plateforme ?](#)

Cas d'usage LIF3

Contexte

Plateforme d'exercices disponibles en compléments des CM_TD_TP de 1ère année à la fac de Lyon.

Utilisée de septembre 2014 à janvier 2015 dans l'UE LIF3 - Prog Fonctionnelle et récursive
120 étudiants au début du semestre, 90 à l'exam

Trace

Actuellement dans 1 BD. Travail à faire pour exporter dans un kTBS.

Traces anonymes (ID user mais aucune autre info).

1 base de trace contient les infos de tous les étudiants

Contient les tables (entre autres) :

- Table **Claire_exercise_item**
 - Id : de l'exercice généré dans la BD
 - Exercise_id : [Id de l'exo](#) pour la plateforme ASKER
 - Type : QCM, QROC, groupement... => pour QCM = multiple-choice
 - Contient : JSON de la forme (pour les QCM) :

```
{“question”:”xxxx”,
“proposition”:[
  {“text”: “la proposition 1”, right:bool(right/false)},
  ...],
“origin-resource”: n° de la ressource utilisée pour générer la question,
“item-type”:”mutiple-choice-question”}
```

- Table **Claire_exercise_attempt**
 - Id : de la tentative dans la BD
 - Exercise_Id : [id de l'exercice](#) auquel est associée la tentative
 - User_Id : [utilisateur](#) ayant fait la tentative
 - Test_attempt_id : NULL tout le temps...
 - Created_at : [date de la forme 20xx-xx-xx hh:mm:ss](#)
 - Position : NULL tout le temps...

- Table **Claire_exercise_answer** (format pour les QCM) :
 - Id : de la réponse dans la DB
 - Item_Id : [n° de la question dans le QCM](#)
 - Attempt_Id : [n° de la tentative pour l'item](#)
 - Contient : JSON de la forme {“content”:[0,1,0,...]}
 - Mark : [score en pourcentage](#)

=> en bleu ce qui varie et qui peut nous intéresser.

En gros, quand l'apprenant demande un exo :

- le système génère un QCM ayant un ou plusieurs **Claire_exercise_item** en fonction du nombre de questions (à partir des "origin-resource")
- le système génère une tentative **Claire_exercise_attempt** avec à chaque fois l'id de l'utilisateur et de l'exo
- lorsque l'apprenant soumet une question (validation question par question), le système génère une **Claire_exercise_answer** avec le contenu de la réponse et son score

Par exemple pour l'exercice 1 et la tentative n°63, la base contient :

Table Claire_exercise_item

- id = 1, exercice_id =1, Content XXX, origin-resource = 6
- id = 2, exercice_id =1, Content YYY, origin-resource = 4
- id = 3, exercice_id =1, Content ZZZ, origin-resource = 12

Table Claire_exercise_attempt

- id = 63, exercice_id =1, user_id = 4356, date = dddd

Table Claire_exercise_answer

- id = 1, item_id =1, Content xxx, mark = 75
- id = 2, item_id =2, Content yyy, mark = 0
- id = 3, item_id =3, Content zzz, mark = 100

Autre exemple pour l'exercice 50 et la tentative n°52, la base contient :

Table Claire_exercise_item

- id = 142, exercice_id =50, Content XXX, origin-resource = 39
- id = 143, exercice_id =50, Content YYY, origin-resource = 3
- id = 144, exercice_id =50, Content ZZZ, origin-resource = 44

Table Claire_exercise_attempt

- id = 52, exercice_id =50, user_id = 1016, date = 9/9/2014

Table Claire_exercise_answer

- pas de réponse soumise

Conclusion : 1 exo (et donc une tentative) est toujours lié à 1 seule personne (généralisé pour lui) mais peut avoir plusieurs réponses tentées.

Scénarios d'analyse LIF3

Sc1 : qualité des QCM

Quelques idées de Nathalie :

- Pour repérer d'éventuelles erreurs dans les QCM : montrer quels sont les distracteurs choisis massivement (seuil réglable) alors qu'ils sont faux. Idem dans l'autre sens : distracteurs presque jamais sélectionnés alors qu'ils sont justes.
- Pour repérer les questions trop faciles ou trop difficiles : taux de réussite moyen au-dessus d'un seuil ou en dessous d'un seuil

Sc1a : Taux de réussite pour une question de QCM

Objectif : connaître le taux de réussite aux différentes questions de QCM qui ont été proposées.

=> pour chaque question, avoir un pourcentage de réussite

Attention :

- pour une même question de QCM, tous les apprenants n'ont pas forcément les mêmes propositions de réponse.
- les apprenants peuvent répondre plusieurs fois à la même questions, avec ou non les mêmes distracteurs
 - => nous ne garderons que la première réponse
- les questions de QCM peuvent avoir plus d'une réponse juste
 - => nous ne considérerons une question comme juste seulement si toutes les bonnes (et uniquement celles-ci) ont été cochées.

Entrée :

- created-at (date)

Action :

- supprimer les données avant le 8/9 et après le 6/01

Résultat :

- mêmes infos dans les traces sur une période plus courte

REMARQUE : en rouge les paramètres à faire varier pour ré-exploiter au printemps...

Entrée :

- les traces précédentes où l'on va regarder
 - les champs "question": "xxxx" de la table **Claire_exercice_item** et **User_Id** de la table **Claire_exercice_attempt**

Action :

- pour chaque couple, question-utilisateur, ne garde que la première instance

- c'est à dire l'ID de la Table `Claire_exercice_attempt` le plus faible

Résultat :

- les mêmes infos mais avec uniquement la première réponse de chaque user aux questions

Entrée :

- les traces précédentes

Action :

- ne garder que les infos sur les questions et leur score i.e. :
 - Table `Claire_exercice_item` - champ "question": "xxxx",
 - Table `Claire_exercice_answer` - champ `Mark` ,
 - Table `Claire_exercice_attempt` - champ `User_Id`

Résultat :

- les questions de QCM sans se soucier des distracteurs et leur score pour chaque apprenant

Entrée :

- les traces précédentes

Action :

- séparer en autant de traces que de questions de QCM

Résultat :

- si n questions, alors n traces avec plusieurs couples score - user

Entrée :

- pour chaque trace créé précédemment

Action :

- faire une moyenne

Résultat = nos indicateurs : pour chaque question de QCM, le taux moyen de réussite à la première tentative.

Sc1b : Taux d'utilisation d'un couple question-distracteurs

Objectif : connaître les distracteurs faux qui sont le plus souvent choisis pour une question données

=> pour chaque couple question-distracteur, avoir un pourcentage d'utilisation

Attention :

- nous n'étudierons pour cet indicateur que les couples questions - réponse fausse

Entrée :

- created-at (date)

Action :

- supprimer les données avant le 8/9 et après le 6/01

Résultat :

- mêmes infos dans les traces sur une période plus courte

REMARQUE : en rouge les paramètres à faire varier pour ré-exploiter au printemps...

Entrée :

- les traces précédentes

Action :

- faire autant de traces que l'on a de couple question-distracteur
 - i.e. dans la table Table **Claire_exercise_item** regarder "question": "xxx" + "proposition": [{"text": "la proposition 1", right: bool(right/false)}, ...]
 - garder la question, la proposition, le right pour savoir si c'est une question juste ou fausse
 - garder aussi dans la table Table **Claire_exercise_answer** le champ Content : JSON de la forme {"content": [0,1,0,...]} pour savoir ce que l'apprenant a répondu

Résultat :

- n traces où une trace contient toutes les instances d'un couple question-distracteurs

Entrée :

- les n traces précédentes

Action :

- supprimer toutes les traces où dans [{"text": "la proposition 1", right: bool(right/false)}, ...] le booléen est vrai

Résultat :

- m traces où une trace contient toutes les instances d'un couple question-distracteurs faux

Entrée :

- les m traces précédentes

Action :

- pour chaque trace, compter le nombre de réponse fausse (i.e.. le nombre de fois où dans content on voit un 1 pour le distracteur (i.e. coché) et calculer le rapport nombre sélection/nombre total

Résultat = nos indicateurs : m taux d'utilisation du distracteur faux pour une question.

Sc2 : Identifier les “types” d’apprenants

Sc2a : Quand les apprenants utilisent-ils la plateforme ?

Objectif : associer 1 ou plusieurs types à un apprenant parmi : utilisateur régulier, occasionnel, avant-TP-noté, avant-exam, au-debut-pas-après...

=> pour chaque apprenant avoir tous les types en sortie avec un booléen

Entrée :

- created-at (date)

Action :

- supprimer les données avant le 8/9 et après le 6/01

Résultat :

- mêmes infos dans les traces sur une période plus courte

REMARQUE : en rouge les paramètres à faire varier pour réexploiter au printemps...

Entrée :

- résultat précédent

Action :

- pour chaque apprenant
- séparer les tentatives par semaine à partir du 8/9
- compter le nombre de tentatives par semaine

Résultat :

- autant de listes que d’apprenants avec autant d’entiers que de semaine entre le 8/9 et le 6/01, ici 18 semaines

Entrée :

- résultat précédent

Action :

- fusionner les semaines 8 (du 27/10) et 9 (du 3/11) car vacances scolaires
- fusionner les semaines entre le 15/12 et le 6/01 car vacances scolaires

Résultat :

- autant de listes que d’apprenants avec 15 semaines donc 15 entiers

=> fin de la préparation “temporelle” des données

Entrée :

- résultat précédent

Action :

- pour chaque nombre de tentative, si nombre de tentatives >1, remplacer l’entier par 1, sinon par 0

Résultat :

- autant de listes que d’apprenants avec 15 entiers valant 0 ou 1

=> fin de la normalisation, début de l’analyse des types d’apprenants

CAS 1

Entrée :

- résultat précédent : pour 1 user, 15 semaines

Action :

- faire la somme de toutes les semaines pour 1 utilisateur
- si somme > 10 alors type régulier = vrai, sinon faux

Résultat :

- pour chaque apprenant un booléen indiquant le type "régulier"

CAS 2

Entrée :

- résultat précédent : pour 1 user, 15 semaines

Action :

- faire la somme de S1 à S7 pour 1 utilisateur => valeur SommeDebut
- faire la somme de S8 à S15 pour 1 utilisateur => valeur SommeFin
- si SommeDebut > 5 et SommeFin < 2 alors type au-début-pas-apres = vrai, sinon faux

Résultat :

- pour chaque apprenant un booléen indiquant le type "au-début-pas-apres"

CAS 3

Entrée :

- résultat précédent : pour 1 user, 15 semaines

Action :

- faire la somme de toutes les semaines pour 1 utilisateur
- si somme <5 et valeur de S9=1 alors type avant-tp-note = vrai, sinon faux

Résultat :

- pour chaque apprenant un booléen indiquant le type "avant-tp-note"

CAS 4

Entrée :

- résultat précédent : pour 1 user, 15 semaines

Action :

- faire la somme de toutes les semaines pour 1 utilisateur
- si somme <5 et valeur de S15=1 alors type avant-exam = vrai, sinon faux

Résultat :

- pour chaque apprenant un booléen indiquant le type "avant-exam"

CAS 5 : type "occasionnel" : 1 semaine sur 2 (ou presque...)

Sc2b : Comment les apprenants utilisent-ils la plateforme (plusieurs essais sur la même version du même exercice) ?

Objectif : savoir si les apprenants font 1 tentative de réponse par exo ou plus

Entrée :

- user-id
- exercise-id

Action :

- compter le nombre N de tentatives par exercice (et donc par apprenant car 1 tentative est le lien entre 1 exo et 1 apprenant)
- si pour chaque exercice, $N = 1$ alors type = "1 tentative"
- si pour chaque exercice, $N > 1$ alors type = "plusieurs à chaque fois"
- sinon, si N est dans l'intervalle $[\text{Nombre d'exo} + \frac{1}{3}; 2 * \text{nombre d'exo} - \frac{1}{3}]$ alors type = "variable"
- sinon "ne sais pas"

Résultat :

- 1 type par apprenant : "1 tentative / plusieurs à chq fois / variable / ne sais pas"

Sc2c : Comment les apprenants utilisent-ils la plateforme (plusieurs versions du même exercice) ?

Objectif : savoir si les apprenants demandent plusieurs versions du même exercice

Entrée :

- user-id
- exercise-id
- origin-resource

... manque des infos dans les traces : on a la ressource dont est issu la question, mais on ne sais pas de quel modèle viennent les exercices :-/

Sc2d : Pourquoi du comment les apprenants utilisent la plateforme ?

Objectif : savoir pourquoi l'apprenant fait plusieurs tentatives de réponse sur une même version de l'exercice.

Entrée :

- user-id
- exercise-id
- answer-item-id
- answer-attempt-id
- mark (une answer-item-id d'une answer-attempt-id peut avoir plusieurs marks)
- résultat précédent

Action :

- combiner les colonnes answer-item-id & answer-attempt-id pour avoir answer-attempt-item-id

Résultat : R1

- une liste avec
 - user-id
 - exercise-id
 - answer-attempt-item-id
 - mark

Entrée :

- résultat précédent

Action :

- faire la moyenne des scores (mark) par answer-attempt-item-id

Résultat : R2

- 1 moyenne par answer-attempt-item-id

Entrée :

- R1 mois mark
- R2

Action :

- pour chaque exercice (et donc chq user) calculer
 - le nombre de réponse
 - le score de la 1ère réponse
 - le score de la dernière réponse
 - le score min
 - le score max
 - la moyenne des scores

Résultat :

- pleins de nombres donc découper tout cela :-)

Analyse : comparer les résultats du Sc2b et Sc2d :

- si Sc2b = "plusieurs tentatives"
et dans Sc2d min < 20 et max > 80
alors type = "pour réussir"
- si Sc2b = "plusieurs tentatives"
et dans Sc2d max < 50
alors type "pas compris"
-