

**Date :** 29/11 - 10h30

**Participants :** Francis Jambon, Philippe Mulhem, Bastien Gattaciecca, Juliette Descotils

**Ce qui a été fait :**

Décisions :

- Il est préférable de renvoyer un fichier après chaque assessment. Chaque document en sortie doit contenir : un time code, l'identifiant de l'expert, du topic, du document ou snippet, un ID pour le type d'assessment et la liste des mots associés à leur pertinence.
- Les niveaux de pertinence sont définis pour chaque campagne d'annotation. Il y a toujours entre 2 et 4 niveaux.
- Le niveau de pertinence par défaut est "not relevant" et tous les mots sont instanciés à "non annoté" au départ. Tous les mots non annotés sont automatiquement passés à "not relevant" lorsque l'expert déclare avoir terminé l'assessment.
- Une tâche atomique correspond à un assessment. On crée un fichier de sortie par assessment. En créer un par tâche serait trop peu (risque de perte en cas de panne trop grave) et un par mot serait compliqué. Par conséquent, on annule le système d'enregistrement qui devient obsolète (vu qu'il a la même granularité que l'envoi des fichiers de sortie).  
ça implique qu'une fois l'assessment commencé il doit être terminé pour être sauvegardé. Donc un assessment se commence et se finit sur une même machine obligatoirement.
- Tableau de bord : un général accessible aux organisateurs uniquement qui regroupe l'avancement des différents experts. Un secondaire par expert leur permettant de voir le nombre d'assessment restant.

**À faire :**

Modifier CDC :

Corriger les définitions du glossaire qui sont trop détaillées et qui rentrent trop dans le cahier des charges, il faut se contenter de définir les mots tels qu'ils sont utilisés dans le domaine

Modifier certaines définitions non comprises (task, document)

Ajouter certaines définitions (SERP, assessment)

Modifier l'ordre des mots du glossaire

Choix :

- où se trouve l'intelligence du système, c'est-à-dire qui choisit quel est l'assessment suivant : le client (client lourd) ou le serveur (client léger)
- format du fichier de sortie : json ,xml, texte
- si on doit annoter à la fois les mots et le document, choisir si on a un ou deux fichiers de sortie. Un fichier serait plus cohérent, deux fichiers serait plus simple.

Faire un diagramme de séquence pour montrer nos choix d'architecture

Modéliser l'interface avec les trois framework JavaScript qu'on propose : Angular, react, vueJS

**Questions :**

Quelle est la différence entre corpus et pool ?

**Prochain RDV :**

Lundi 06/12/2021