Date: 15/11 - 10h30

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

## Ce qui a été fait :

Discussion à propos de la technologie utilisée

Discussion sur l'architecture : s'il s'agit d'un standalone (application en client lourd installée sur le pc en local) ou d'une appli Web-based communicant avec un serveur.

Globalement, la contrainte principale est la disponibilité de l'application à distance et le fait que les résultats soient regroupés.

La demande sur la robustesse du code a été reformulée (fonctionne quel que soit l'OS ou le navigateur selon l'architecture choisie, mais dans une moindre mesure : Windows 10 et Firefox respectivement).

Discussion sur l'atomicité de l'enregistrement des données quand l'expert utilise l'interface : est-ce qu'on décide d'enregistrer après chaque annotation, chaque topic, chaque user ?

À priori on préfère l'idée d'enregistrer après chaque annotation.

Discussion sur la reprise après une panne transitoire et définitive.

Précisions à inclure dans le cahier des charges :

- À priori le temps de remplissage est inconnu mais on peut l'estimer à partir du nombre de mots par snippet (25 environ sur un snippet type google) en supposant que l'expert met une seconde par mot. Pour un document, on suppose que l'annotation prend 30 secondes.
- On n'a pas spécialement besoin de récupérer les données dans une base de données, un fichier texte est suffisant.
- Il faut pouvoir gérer les accès concurrents au système.

## À faire :

Définir clairement les termes employés et travailler sur le glossaire dans le cahier des charges.

Choisir si une tâche correspond à l'ensemble des mots et au snippet total ou si on sépare en deux tâches, une pour les mots du snippet et une pour le snippet dans sa totalité.

Choisir comment gérer la sauvegarde des données (en local après chaque opération, pratique et rapide pour les pannes transitoires et notification au serveur toutes les 10s par exemple) et déterminer sa structure.

Choisir une méthode pour annoter la pertinence des mots (plusieurs solutions : cliquer plusieurs fois, clic gauche qui incrémente de 1 et clic droit qui incrémente au max, clic glissé, ...)

## Questions:

Comment enregistrer les données après une panne transitoire ? définitive ?

D'autres spécifications de besoin pour le cahier des charges ?

comment on gère les pannes = si pb réseau, on récupère les données ou pas ? ça peut générer des pb. si on renvoie une page vide c'est peut-être plus simple même si enft 2 types de panne :

pb réseau = granularité d'enregistrement au topic, si pas d'enregistrement tant pis, quand on reload on a perdu les données est-ce que on fonctionne en boucle local = va vite mais on perd tout si bug

boucle avec serveur = + long mais sauve données en continu

reprise sur échec n'est pas vitale mais ne pas faire bosser les gens pour rien non plus, afficher immédiatement si panne reprise sur panne transitoire mais pas sur panne définitive (gpacompri)

au niv du système on veut un enregistrement par tâche au niv dev il faut qu'on voit

si on enregistre en local, on peut éviter pb de panne réseau temporaire (faisable en standalone et en web based)

tableau de bord : synthèse à chaque enregistrement? c quoi le tablo de bord ?

choix de techno

gérer le projet par les risques ? identifier les difficultés techniques, risques = comment on fait pour annoter un mot (difficulté de le gérer donc risque d'échec)

trois niveaux de pertinence : pertinent topical non pertinent cliquer plusieurs fois c un peu embêtante mais mieux que cliquer et se déplacer option clic gauche vs clic droit clic glissé

tout le monde doit pouvoir l'utiliser facilement

## à faire :

bien définir les mots commencer à réfléchir à la structure de données de l'enregistrement → aller au + simple historique ajoute exemple relevation contrainte matérielle = multiplateforme autre contrainte robustesse + évolutivité