

Semaine 1.2

Prof absent : Mme Bahsoun

Elèves absents : Rania (pour ses dents), Aissa (justification donnée à Mme Chouquet), Amine (Avion), Jeanne (malade), Marina (Malade : arrête maladie pour la semaine donnée par le médecin)

Groupe 1

Mettre la BD sur le serveur (en anglais)

Décider la licence pour le code (voir avec le Groupe 9)

Groupe 2

Objectifs du groupe pour la journée atteints et dépassés

Procédures ok (il en reste 2)

Déclencheurs ok (il en reste quelques un pour demain)

MCD validée

API fonctionnelle (quand on insert un article on arrive à obtenir des id)

Finaliser les procédures et les déclencheurs, lancer la BD sur le serveur, faire des tests, commencer/terminer les vues matérialisées

Groupe 3

Récupération des requêtes du groupe web, script testé en local (normalement ok) – à tester sur le serveur

Recherche sur les vues matérialisées

Réflexion sur les requêtes et la mise en forme des fichiers (pour le Groupe 8)

L'API fonctionnelle, il faut tout passer en flask (en cours, problème pour trouver la bonne adresse)

Essayer de faire fonctionner l'API (trouver l'adresse)

Groupe 4

Fin Futurascience, le Figaro, 20 minutes, Nouvelle Obs (manque les anciens), Libération, La Dépêche (manque les anciens)

Reste à faire le dé-doublonnage, nettoyage du code et mise sur le serveur

Groupe 5

Premier jet pas très propre

Lemmatisation réalisée (juste tronquée, donc pas finie) et donnée au Groupe 6

Travail sur une meilleure lemmatisation (reconnaissance d'entité nommée, stop-words) + prise de contact avec les autres groupes concernés (post-taggings et mise en page des fichiers à envoyer)

Dé but de fonction pour le post-tagging

Mettre au clair les fonctions expérimentales qui fonctionnent

La lemmatisation doit se poser sur le post-taggings

Faire attention aux accolades (Groupe 5) parce que sinon il peut y avoir des soucis avec l'API + format

Groupe 6

Plus de scrappage de Wikipedia, utilisation du package Wikipedia (qui utilise directement la barre de recherche de Wikipedia) : fonctionnelle

Travail sur les articles « grave » ou « pas » : à partir d'une liste de mots classifiés comme grave ou pas, déterminer si c'est grave ou pas

Optimisation du code (réduction par 40 du temps de traitement)

Utilisation d'un package pyquery qui permet d'avoir accès à tous les barres de recherche (utilisation de Synonymo pour récupérer les synonymes)

Enlever les mots neutres du traitement

Groupe 7

Recodage de la totalité des articles fini

Groupes sur des listes de méthodes, pré-processing des textes pour les mettre dans les algos (pareil pour les bag of words)

Faire tourner les algos et les optimiser

Groupe 8

Préparer les mises en forme des fichiers à recevoir (json)

Analyses de tendance (création des codes par jour et par mois)

Statistiques à la journée + tests statistiques pour les tendances : OK

Revue de code en cours

Besoin de données (même fictives) pour tester les codes et d'optimiser

Séries chronologiques

Groupe 9

Structure du site web (bootstrap)

Développement du site sans utiliser les données, bon affichage du site

Problème de communication pour le transfert des données, entre le Groupe 2 et le Groupe 9

Récupération des fichiers json et donc les utiliser pour dynamiser le site

Groupe 10

Charte de codage OK

Charte graphique OK

Liste des licences faites, *Groupe Supervision doit décider maintenant la quelle choisir (réunion avec Groupe supervision demain)*

Template des rapports OK

Css et tests