Université Claude Bernard Lyon 1

Dossier d'initialisation (D0)

Collection and Analysis of Tweets made Simple

Master2 TI-Promo 2015
Abdelkader Messeguem
Selim Dogan
Cihan Cinar
Hicham Boubouh
Nathanaël Desjobert

Encadrant : Fabien Rico
Chef d'équipe : Anthony Deseille

15 octobre 2015

Sommaire

1 Objet et contexte :

1.1 Contexte :

CATS (Collection and Analysis of Tweets made Simple) est une plate-forme d'analyse et de récolte de tweets développé en Python. Cette application est la suite d'un Travail d'Études et de Recherche mené par Monsieur Rico, notre porteur de projet. Elle a été réalisée en partenariat avec des doctorants de l'Université Lumières Lyon 2. Le but de CATS est de permettre aux utilisateurs de collecter facilement des tweets en fonctions de certains critères ou filtres. Les filtres peuvent être multiples : mots-clés, localisation, utilisateurs, dates. De plus, CATS met en oeuvre différents outils d'analyse de tweets tels que la détection d'évènements, la reconnaissance d'entités, et la visualisation en nuage de mots.

1.2 Positionnement :

Actuellement, l'application fonctionne partiellement et a été créée d'une manière simple et rapide. Ainsi, il est difficile de faire des maintenances ou des évolutions telle que l'ajout de nouveaux modules. C'est pourquoi, l'équipe de CATS a décidé de faire appel à nous afin que nous reprenions l'ensemble du projet pour qu'il soit générique, documenté, et maintenable. L'objectif principal dans ce projet est donc la création d'une version stable de CATS.

1.3 Objet :

Dans un premier temps, l'objectif du projet est le développement d'une version avec les fonctionnalités de bases suivantes :

- Collecte d'un ensemble de tweets dans une durée donnée avec l'utilisation d'au moins un filtre (mots clé, géolocalisation, comptes utilisateurs).
- Stockage du corpus de tweets dans une base de données. Initialement, nous stockerons les informations suivantes en base : la date de publication, le tweet, le pseudo, le vrai nom de l'auteur, la description de son profil, la langue du tweet donnée par Twitter, et enfin, le type du tweet (original ou retweet). Cette liste d'attributs pourra être modifié par la suite.
- Marquage automatique du corpus. C'est à dire, créer ou utiliser un système de détection du sexe avec le prénom de l'auteur.
- Interrogation du corpus avec différents filtres.
- Export et import de corpus en CSV

Par la suite, le projet devra implémenter des fonctions plus avancées telles que :

- Pour chaque tweets, l'application doit être capable de repérer les entités nommées et leur assigner un type (lieu, organisation, personne) et si possible déterminer leurs probabilités. Il est aussi demandé d'avoir une visualisation lemmatisée du tweet et de connaître chaque utilisateur mentionné.
- Pour le corpus, l'application doit fournir le vocabulaire intégral, ainsi que toutes les personnes nommées et toutes les entités nommées avec pour chacune la fréquence.
- La méthode de filtrage doit être plus évolué et ergonomique via des expressions régulières, utilisable par des linguistes. La méthode consiste à extraire tous les mots "commençant par…" ou/et "finissant par" et à extraire tous les tweets contenant une expression régulières.
- Le programme doit être totalement générique dans l'analyse de tweets et de corpus ainsi que dans la méthode de filtrage. C'est à dire qu'il doit exister la possibilité d'importer des modules externes au projet.

Enfin, toutes les fonctionnalités doivent être accessibles depuis une interface web qui doit être ergonomique. Cette interface doit permettre :

- La gestion d'un compte utilisateur
- Le lancement d'une collecte
- La visualisation d'un sous corpus
- De visualiser les résultats du vocabulaire intégral, les entités et les personnes sous forme de nuage
- La visualisation ou non sous forme lématisée
- Le téléchargement d'un corpus ou sous corpus créé avec des filtres
- L'utilisation d'un feedback pour la vérification des robots et la vérification des marquages automatiques.
- L'affichage du pourcentage d'erreur des marquages et présence de robots.

2 Résultats attendus :

La liste des livrables qui seront produits.
Livrables de gestion de projet :

- 26/10/2015 - GdP Post-sprint 1
- 07/12/2015 - GdP Post-sprint 2
- 08/02/2016 - GdP Post-sprint 3

Livrables techniques :

- Pré-sprint 1 — 28/09/2015
    1. Cahier des charges
    2. Conception / Modélisation
    3. Maquettes de l'interface
- Post-sprint 1
    - 19/10/2015 — Application minimale :
        1. Authentification sur l'application avec un compte Twitter.
        2. Possibilité de lancer la collecte de tweets avec un ou plusieurs filtres
        3. Stockage de corpus en base de données avec la possibilité de l'exporter.
    - 26/10/2015 : Remise du bilan du Sprint 1
- Post-sprint 2
    - 30/11/2015 — Application amélioré :
        1. Ajout de fonctionnalités supplémentaires pour l'analyse de corpus, ainsi que pour les méthodes de filtrages.
        2. Amélioration de l'interface web (visualisation des résultats, Lemmatisation des tweets)
    - 07/12/2015 : Remise du bilan du Sprint 2
- Post-sprint 3
    - 01/02/2016 — Application final :
        1. L'application devra être implémenté au mieux avec toutes les fonctionnalités décrites dans le cahier des charges.

- 08/02/2016 — Remise bilan et documentations :
  1. Remise du bilan du Sprint 3
  2. Documentation technique (type Javadoc)
  3. Documentation d'installation
  4. Compte rendu technique (temps d'exécution, temps de téléchargement, espace disque occupé)
  5. Documentation de développement (description des templates pour les filtres et le marquage)

Autres livrables :

- 16/02/2016 — Support de présentation de soutenance

3 Méthode et outils :

3.1 Contraintes

Contraintes de délais:

- L'application devra être fonctionnelle avec toutes les fonctionnalités demandés à la fin de l'UE TI5 Projet.

Contraintes de coûts:

- Nous serons une équipe de 6 personnes sur le projet pour un total de 558h de travail.

Contraintes de qualité:

- Une documentation et des tests devront être réalisés à chaque fin de sprint. Une version stable et bien documenté sera apprécié pour la reprise du projet.

3.2 Méthode :

La gestion du projet se fera avec une méthode Agile nommée "Scrum". Cette méthode s'appuie sur le découpage d'un projet en boîte de temps nommées "sprints". Pour ce projet, nous aurons 3 sprints pendant lesquels nous pourrons réaliser les objectifs que nous nous sommes fixés. Chaque sprint se terminera d'une démonstration de ce qui a été achevé.

3.3 Outils :

3.3.1 Détails technique

Au niveau des détails techniques, nous avons décidé d'utiliser les technologies suivantes :

- Framework J2EE : Spring
- Base de données : Lucene
- Collecte de tweets : Librairie en JAVA "Streaming API Twitter"
- Nuage de mots : WordCram
- Framework Front-end : JSP et Normalize

3.3.2 Gestion de projet

Pour une une bonne communication entre les membres de l'équipe, nous avons mis en place une application de messagerie Slack. Cet outil, nous permettra de partager les heures de réunions, les liens utiles, les documents mis en place mais aussi de réfléchir sur des points techniques. De plus, la forge de l'Université sera également mis en place pour le dépôt du projet et l'assignation et la visualisation de l'avancement des tâches. Enfin, pour les documents à rédiger, un drive Google a été mis en place.

4 Macro-planning:

4.1 Lots de travail :

4.1.1 Général

- Documentation : Rédaction d'une documentation (type Javadoc) tout le long du développement
- Tests : Des tests unitaires seront créé à chaque nouvelle fonctionnalité

4.1.2 Back-End

- Mise en place de la base de données : Création de la base de données qui stockera le corpus, ainsi que la mise en place du mapping en Java.
- Authentification avec un compte Twitter : Authentification sur l'application avec un compte Twitter pour récupérer les clés pour l'utilisation des différentes APIs
- Mise en place de la collecte (récupération, traitement, stockage) : Implémentation de la Streaming API pour la collecte et implémentation du traitement pour chaque tweet (par exemple, marquage du sexe de l'auteur).
- Manipulation de la base de données : Développer le fait de pouvoir récupérer des informations en sous corpus avec des filtres.
- Fonction avancée pour les tweets : Implémentation des fonctionnalités avancées pour les tweets.
- Fonction avancée pour le corpus : Implémentation des fonctionnalités avancées pour le corpus.
- Fonction avancée pour le filtrage : Implémentation des fonctionnalités avancées pour la méthode de filtrage.
- Mise en place d'un feedback : Implémentation du feedback pour les utilisateurs.

4.1.3 Front-End

- Interface de connexion
- Interface création de collecte
- Interface visualisation des résultats
- Interface du feedback
- Export en CSV

4.2 Phasage:

	Sprint 1					Sprint 2					Sprint 3				
Back end															
Mise en place de la base de données	✓														
Authentification avec un compte Twitter	✓														
Mise en place de la collecte (récupération, traitement, stockage)	✓	✓	✓	✓	✓										
Manipulation de la base de données	✓	✓	✓	✓	✓										
Fonction avancée des tweets						✓	✓	✓	✓	✓					
Fonction avancée du corpus						✓	✓	✓	✓	✓					
Fonction avancée du filtrage											✓	✓	✓	✓	✓
Mise en place d'un feedback											✓	✓	✓	✓	✓
Front end															
Interface de connexion	✓	✓													
Interface pour la création de collectes	✓	✓	✓												
Interface de visualisation des résultats			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Interface du feedback											✓	✓	✓	✓	✓
Export en CSV				✓	✓										
Général															
Documentations	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Tests	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓