Master INFORMATIQUE

Analyse de Comportements avec Twitter

2015-2016

Git Repository: `https://github.com/LatifaBouchouaf/Pje_Twitter.git`

Malik Boukrou

Latifa Bouchouaf

Table des matières

1 Introduction

Lors de notre premier semestre de Master 1 Informatique, nous avons eu la possibilité de choisir un projet encadré traitant de l'analyse de comportements sur Twitter. Nous avons opté pour une implémentation des algorithmes et modèles en Java.

Twitter est un célèbre réseau social qui repose sur un système de publication de messages courts. Ainsi, une personne inscrite peut envoyer un tweet à tous ses abonnés, qui peuvent éventuellement le publier de nouveau sur leur propre compte. On peut également s'abonner fil de discution des autres utilisateurs. Twitter nous permet l'accès à une immense source d'informations diverses et variées sous un format réduit. Le but est maintenant de pouvoir extraire automatiquement de ces tweets une information. Nous voulons pouvoir les classifier selon le degré d'émotion qu'ils véhiculent.

Pour ce faire, nous avons d'abord récupérer et stoker les tweets. Pour ensuite implémenter différents algorithmes de classification étudiés en cours. Et pour finir par mettre en place une interface pour traduire ces résultats de manière à ce qu'un utilisateur puisse facilement comprendre aux mieux les résultats.

2 Description générale du projet

2.1 Description de la problématique

Le but de ce projet est de mettre en place un programme capable de traiter un message afin de pouvoir en extraire une information qui peut le classifier. Et il faut aussi pouvoir gérer tout un flux de message. On peut ici pouvoir les classer en fonction du sentiment véhiculé par le message. Bien sûr, cette tache est beaucoup mieux réalisé par un être humain, mais on essaye ici d'être le plus fidèle possible.

2.2 Description générale de l'architecture de votre application

Notre projet comporte 4 packages pour correspondre au mieux au model MVC. Il y as :

— classification qui contient les classes des algorithmes de classification.

— controleur qui contient les classes des actions des différents boutons.

— model qui contient le main de l'application ainsi que le fichier de configuration d'accès au compte twitter, les fonctions de récupérations des tweets et les outils statistiques.

— view qui les vues des différents calculs statistique et de l'interface.

3 Détails des différents travaux réalisés

3.1 L'API Twitter

Pour ce projet, l'API utilisé est Twitter4j. Elle nous permet l'envoi de tweets ainsi que la récupération à condition de bien le configurer. Ici, nous n'utiliserons que la récupération. Tout les paramètres, que ce soit le consumerKey jusqu'au port, sont enregistrer dans le fichier AppliSettings.

3.2 Préparation de la base d'apprentissage

Une fois récupérés, les tweets nécessitent un nettoyage et ensuite être stockés.

3.2.1 Netoyage des données

Pour chaque tweet sauvegardé dans la base, nous avons également enregistré l'identifiant, l'auteur, le texte nettoyé, la date de création et la valeur de la classe du tweet. Pour pouvoir analyser au mieux les messages, il est important de les nettoyer afin d'éviter au maximum le bruit dû aux différentes syntaxes propre à twitter ou simplement due à la forme du message. On a pour ce faire supprimer tous les URL, RT, hashtag, les nombres, etc. À la fin du nettoyage, il ne reste seulement que les mots qui conservent leur ordre et les smileys.

3.2.2 Construction de la base

Pour chaque tweet sauvegardé dans la base, nous avons également enregistré l'identifiant, l'auteur, le texte nettoyé, la date de création et la valeur de la classe du tweet. Toutes ces informations sont écrites dans un fichier CSV, comme illustré par la figure 1.

```
676425978495676416;Ash Ketchum;pipika ot C on Pikachu;Mon Dec 14 16:38:15 CET 2015;2
676425382791389184;Camille MP13;petit ikachu ouais;Mon Dec 14 16:35:53 CET 2015;2
676423925652430848;Mahn Listen.;DShawXX plus pikachu;Mon Dec 14 16:30:05 CET 2015;2
```

FIGURE 1 – Extrait de la base.

3.3 Algorithme de classification

Les algorithmes implémentés sont des algorithmes de classification supervisés. Ils ont pour but de deviner l'appartenance d'un objet à une classe définie en fonction des objets qui la constitue en fonction d'une certaine similarité.

3.3.1 mots clefs

L'algorithme de recherche par mots clé est un algorithme qui va tout simplement classer un tweet en choisissant la classe qui comporte le plus de mots en commun avec le tweet. C'est un algorithme naïf pour lequel nous avons utilisé deux classes de données qui sont constituées pour l'une de mot positif et pour l'autre de mot négatif. Si le nombre de mots négatif est égal au nombre de mots positif, le tweet est considéré comme étant neutre.

3.3.2 KNN

KNN est la méthode des plus proche voisin. Elle vient calculer la distance entre le tweet et ses n plus proches voisins. Elle attribut la classe la plus présente parmi les tweets d'un échantillon déjà étiqueté. Pour calculer la distance entre un tweet et sont voisin, il faut appliquer l'équation suivante :

$$\frac{NbTotal - NbMot}{NbTotal}$$

Où NbTotal est le nombre total de mots des deux tweets et NbMot est le nombre de mots en commun. Pour cet algorithme, plus l'échantillon est grand est plus le résultat est fiable. Dans notre cas, nous avons utilisé les 30 plus proches voisin de la base de tweet.

3.3.3 Bayes

Bayes est un algorithme qui va calculer la probabilité qu'un tweet fasse partie d'une classe en fonction des éléments qui en font partit. Cette méthode se traduit par : où t représente le tweet, c représente la classe et m représente un mot du tweet.

$$P(t|c) = \prod_{m \in t} P(m|c)$$

où t représente le tweet, c représente la classe et m représente un mot du tweet.

Cette méthode à plusieurs variantes possible. Tout d'abord, on peut choisir de prendre en compte la fréquence d'apparition d'un mot dans le tweet. On aura donc l'équation suivant :

$$P(t|c) = \prod_{m \in t} P(m|c)^{n_m}$$

où n_m est la fréquence du mot m.

La deuxième variante consiste à ne plus à tester un mot à la fois, mais un groupe de mots. Car on part du principe que certain mot ont une plus forte importance lorsqu'il est associé à d'autre mot comme par exemple ras de marée.

3.4 Interface graphique

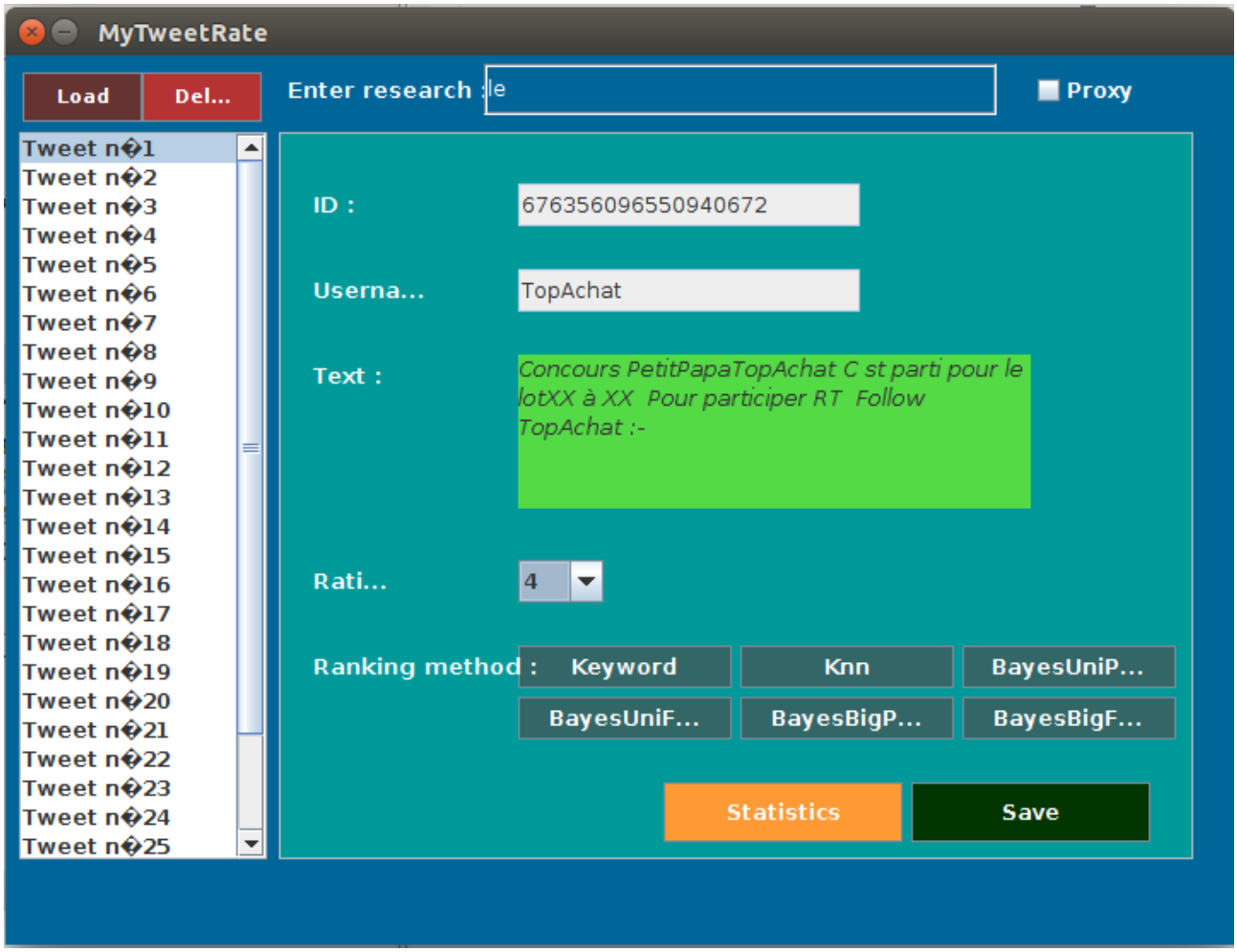

FIGURE 2 – Capture d'écran de l'interface.

Sur la figure 2, nous avons une vue d'ensemble de l'interface. Elle est constituée de plusieurs zones bien distinctes.

Pour commencer dans le haut de gauche à droite. On peut trouver un bouton Load, il permet de recharger les tweets sauvegardés dans la base de tweets.Ensuite, on voit le bouton Delete, qui sert à vider la base de données et qui affiche une fenêtre de confirmation comme sur la figure 3. Puis nous avons la zone de saisie de texte Enter researche, elle permet de récupérer les nouveaux tweets présent sur le compte twitter et les affiches avec les tweets présent dans la base de tweets. Et toute à droite, vous avez la possibilité d'utiliser le proxy que vous avez configuré dans le fichier de configuration en cochant la case.

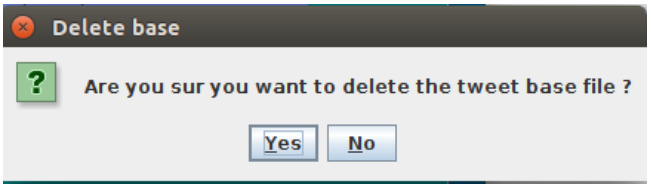

FIGURE 3 – Confirmation de l'effacement de la base de tweet.

On peut voir à gauche une liste de clicable de tweets, il suffit de cliquer sur le tweet voulu pour afficher toutes ses informations sur la droite. On peut y retrouver l'Id, le nom de l'auteur et le texte du tweet. La zone qui affiche le texte du tweet peut s'afficher de différentes couleurs. Si le tweet est positif le fond est vert, s'il est négatif le fond est rouge, s'il est neutre le fond est bleu et s'il n'est pas encore annoté le fond est gris.

Et nous avons aussi une liste de choix Rate qui nous permet d'annoter manuellement le tweet sélectionné.

Le bouton Save vous permet de sauvegarder les nouveaux tweets venant du compte twitter et de mettre à jour les valeurs des tweets déjà présent dans la base de données. À chaque click, une fenêtre comme celle de la figure 4 vous demande confirmation.

FIGURE 4 – Confirmation de sauvegarde.

Et pour finir, le bouton Statistics qui vous permet d'afficher les ratios des différentes classes parmi les tweets et les taux d'erreur de chacun des algorithmes, comme illustré dans la figure 5.

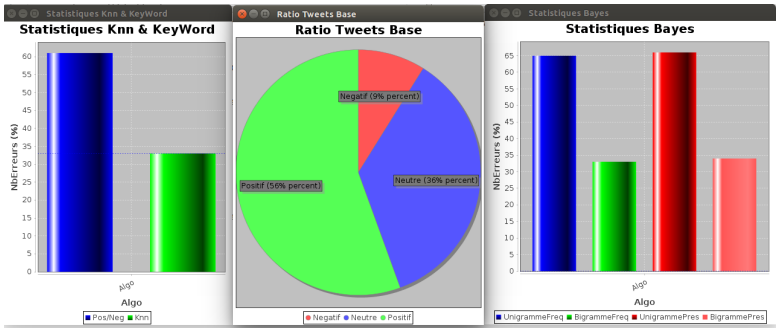

FIGURE 5 – Affichage des statistiques.

4 Résultats de la classification et analyse

À l'aide de la figure 5, on peut constater que les méthodes des mots-clés et de Bayes avec les unigrammes (celle qui ne teste qu'un mot) ont un taux d'erreur de l'ordre de 60% alors que les méthodes de KNN et de Bayes BiGramme (celle qui teste deux mots) sont de l'ordre de 30%. Ces performances produisent plusieurs points importants dans les algorithmes de classification.

Pour le KNN, nous avons un bon taux d'erreur puisque notre échantillon de tweets de référence est assez important pour réduire l'impact des faux positif.

Pour les algorithmes de Bayes, on peut remarquer que les versions qui ne testent qu'un mot à la fois sont moins précises que les versions qui testent des groupes de deux mots puisque ce groupement de deux mots vient ajouter une information supplémentaire pour la classification des tweets.

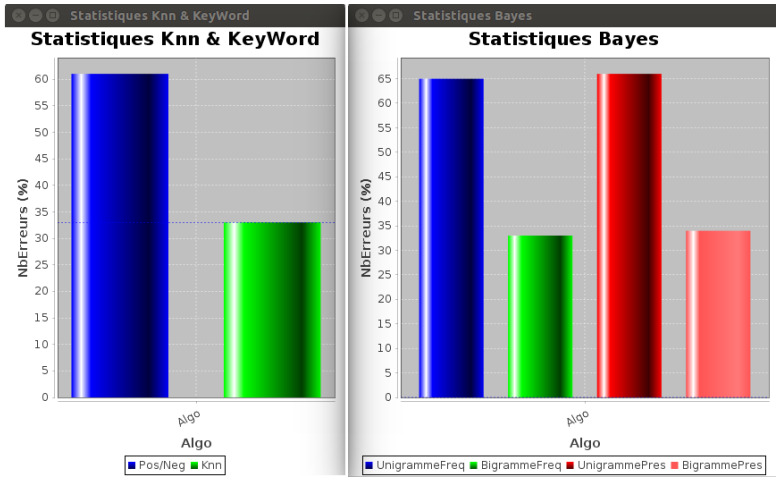

FIGURE 6 – Exemple de résultat de performance.

5 Conclusions

En conclusion, les algorithmes de classifications supervisées que nous avons étudiés sont beaucoup plus fiable lorsque les classes de références sont bien fournie comme on as pu le constater la méthode du KNN. De plus, comme on peut le remarquer avec la méthode de Bayes, il existe beaucoup de point sur les quels nous nous pouvons travailler pour les rendre encore plus performant.

Ce projet nous a aussi permis de développer au mieux nos capacités de travail en binôme, de communication et de recherche d'outils nous permettant de mener à bien notre travail. Nous avons pris conscience de l'importance de l'organisation nécessaire à un projet d'une telle importance.