PJE Twitter - Analyse comportementale

Franquenouille Kevin
Cornette Damien
SVN Repository : `https://svn-etu.fil.univ-lille1.fr/svn/pje14-15-cornette`

Année universitaire : 2014-2015

Introduction

Dans le cadre de notre formation en Master 1 Informatique à Lille 1, nous avons eu la chance de réaliser un projet en Java qui permet de faire une analyse comportementale sur Twitter. Pour cela, nous avons donc implémenter différents algorithmes et modèles.

L'analyse sentimentale sur Twitter est donc devenue un nouveau défi pour le traitement automatique des langages. Cela se répercute sur l'ensemble des réseaux sociaux. Profitant de la qualité et quantité d'information disponible, il y a donc une volonté d'analyse automatique des avis exprimés sur les textes.

Même si le but principal est d'être capable d'analyser les sentiments et les émotions exprimés dans les textes (les tweets dans notre cas) grâce à des algorithmes, la classification ne pourra jamais être meilleure qu'un humain qui classe lui même. Ce dernier permet de détecter les cas ironiques tandis qu'un algorithme ne le permet pas.

Dans ce document, nous allons d'abord commencer avec la description générale du projet avec la problématique, l'API Twitter l'architecture de l'application et l'interface graphique. Dans un second temps, nous étudierons l'analyse et les différentes classifications (KNN, Bayésienne et mots-clés). Pour finir, nous verrons un comparatif global de tous les algorithmes.

Table des matières

1 Description générale du projet

1.1 Problématique

Le principal objectif du projet est de pouvoir réaliser des analyses sur les différents tweets et de pouvoir les classer en fonctions du sentiments qu'il dégage (positif, négatif ou neutre)

1.2 API Twitter

Concernant l'API Twitter, il suffit d'utiliser celle de Twitter4j et d'utiliser le fichier *.jar* disponible.

Pour le bon fonctionnement de notre application, il est essentiel d'utiliser les paramètres de configuration et de connexion afin que l'on puisse récupérer les derniers tweets.

1.3 Architecture de l'application

L'architecture du projet est assez simple. Nous avons choisi d'utiliser le modèle MVC. Voici donc la liste des différents packages :

- *twitter* : contenant le main de l'application
- *parser* : contenant les classes permettant le nettoyage des tweets
- *observer* : utilisé pour mettre à jour la vue
- *model* : contenant tous les modèles et la connexion à l'API
- *front* : contenant tous les composants graphiques de l'application
- *controleur* : contenant toutes les actions sur les boutons et les champs de saisie
- *constant* : contenant une classe regroupant toutes les chaînes de caractères et variables globales
- *classification* : contenant les différentes classification (naïf, KNN et bayésienne)

2 Mode d'emploi

2.1 Lancement de l'application

La première chose à faire est de lancer l'application, pour cela, rien de plus simple, il suffit d'entrer cette commande dans le terminal :

java -jar twitter.jar

Au lancement de l'application, nous arrivons sur cette page vierge de données

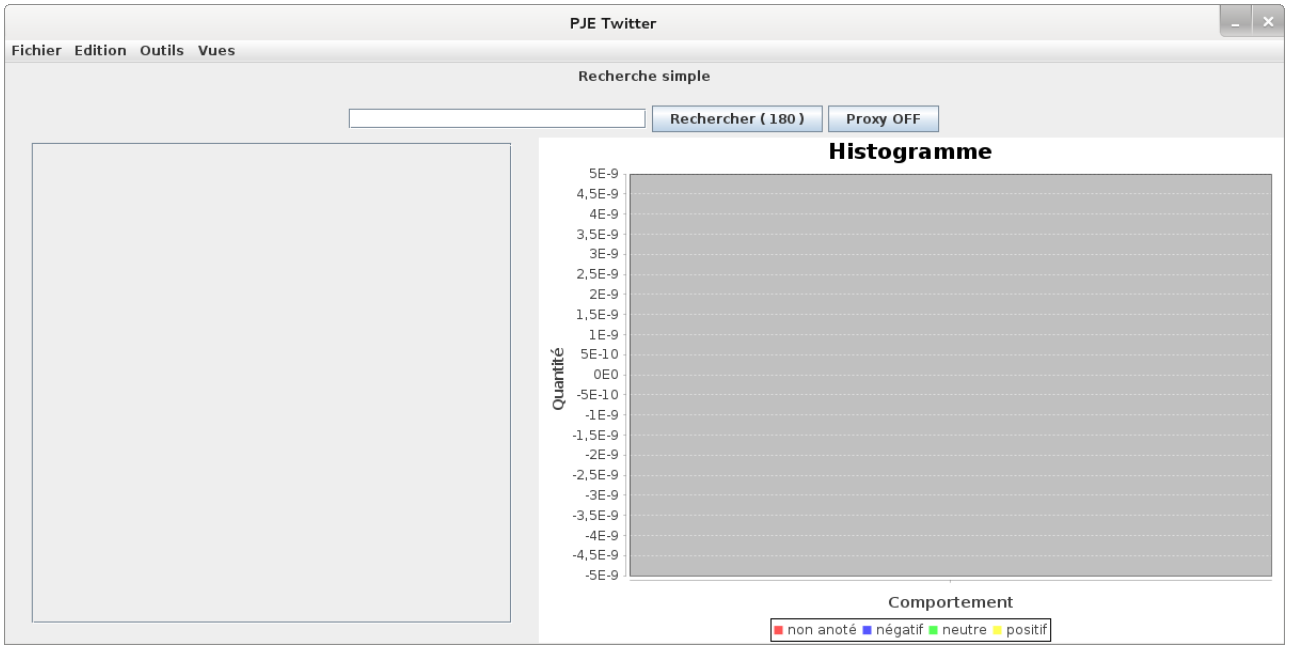


FIGURE 1 – Ecran de démarrage

On y trouve un menu de navigation comportant :

- Un onglet *Fichier* permettant de sauvegarder les tweets et quitter l'application
- Un onglet *Edition* permettant d'accéder aux préférences utilisateurs de l'application
- Un onglet *Outils* permettant de classifier les tweets
- Un onglet *Vues* permettant de choisir le mode graphique (Histograme ou Camember) des sentiments des tweets

On y trouve également une barre de recherche.

FIGURE 2 – Recherche de Tweets

Celle-ci comporte :

- Un champ permettant de saisir le critère de recherche (l'appui sur la touche entrée lance la recherche)
- Un bouton rechercher (ayant le même comportement que l'appui sur la touche entrée sur le champ de saisi)
- Un bouton permettant d'activer ou désactiver le proxy

Enfin, cette page comporte le résultat de la recherche de tweets sous forme de liste à gauche et sous forme graphique montrant la répartition des sentiments à droite.

2.2 Simple recherche de tweets

Avant de classifier des tweets, nous devons d'abord faire une recherche.

Saisissons *babylone* par exemple dans la zone de saisie (Voir Figure 2) et appuyer sur la touche entrée ou sur le bouton *Rechercher* pour valider.

Vous devriez visualiser ce résultat :

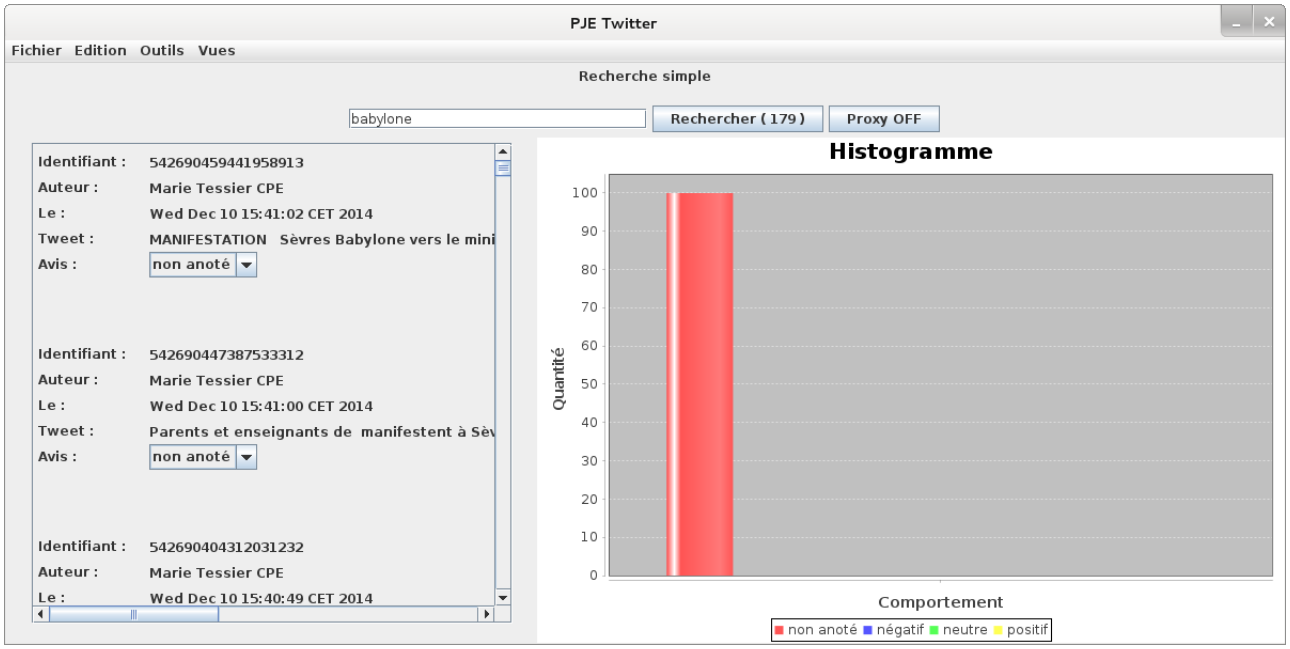


FIGURE 3 – Recherche de Tweets sur le mot *babylone*

Le premier constat qu'on peut faire, c'est que le nombre sur le bouton *Rechercher* vient de diminuer. Ce nombre correspond au nombre de requêtes disponibles vers la base de données Twitter. Initialement, on a le droit à 180 requêtes.

Ensuite, nous pouvons visualiser la liste des 100 tweets les plus récents sur le mot *babylone* à gauche. Chaque tweet est non annoté par défaut, on peut d'ailleurs facilement le vérifier sur le graphique à droite.

2.3 Sauvegarde des tweets

Une fonctionnalité bien pratique est de pouvoir enregistrer notre recherche de tweets dans un fichier CSV.

Pour ce faire, cliquons sur l'onglet *Fichier* de la barre de menu puis sur *Sauvegarder les tweets*. (Notons au passage que le bouton *Quitter* permet bien évidemment de quitter l'application)

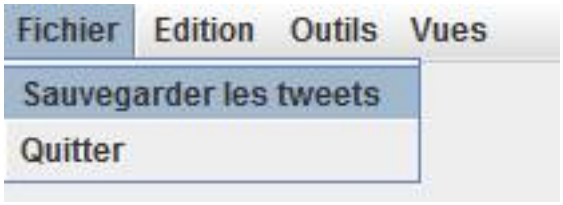


FIGURE 4 – Click sur *Sauvegarder les tweets*

Un écran se propose alors de choisir un nom de fichier (sans l'extension) et sauvegarder les tweets dans ce fichier.

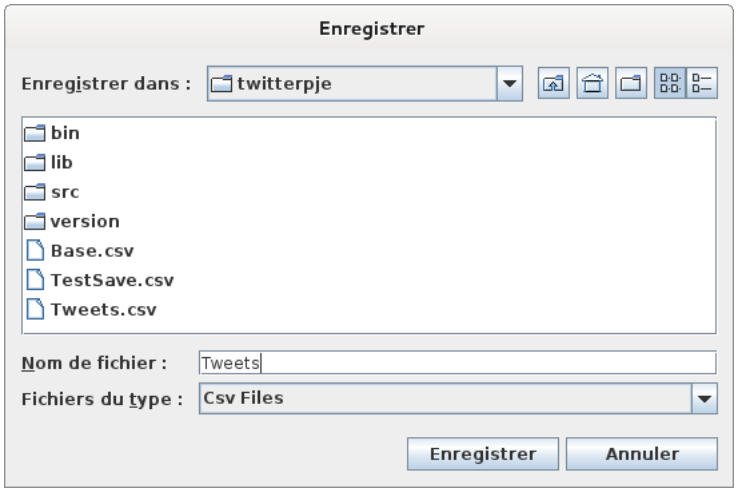


FIGURE 5 – Ecran de sauvegarde des tweets

A la suite d'une sauvegarde réussie, vous recevez ce message :

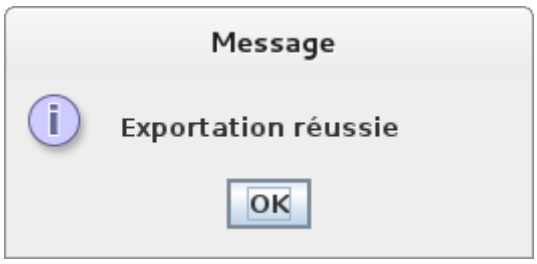


FIGURE 6 – Sauvegarde réussie

2.4 Configuration Proxy

Si vous vous trouvez sur un réseau utilisant un proxy (Exemple : l'université Lille 1), vous aurez besoin de configurer ce proxy sur l'application.

La configuration de ce dernier se passe dans l'espace des préférences utilisateurs.

Cliquons sur l'onglet *Edition* puis *Préférences*

FIGURE 7 – Click sur *Préférences*

Le panneau des préférences s'ouvre :

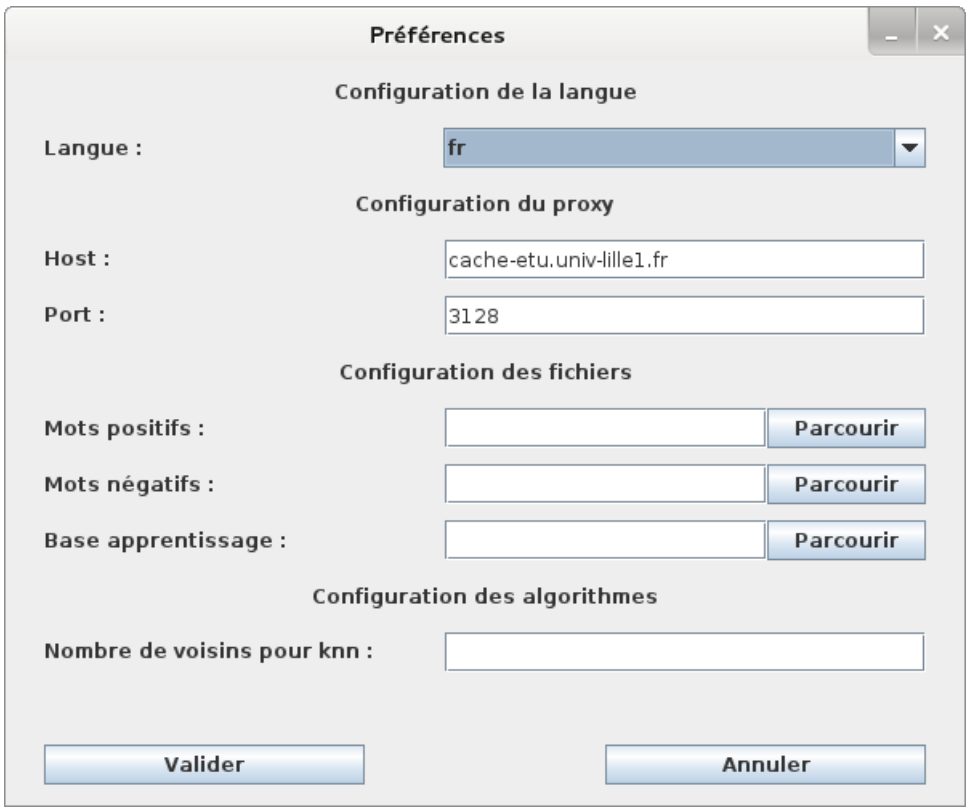


FIGURE 8 – Panneau des Préférences

Dans la section *Configuration du proxy*, il suffit de renseigner les champs *Host* et *Port* (Par défaut, l'application est configuré pour l'université Lille 1) puis de cliquer sur le bouton *Valider*.

De retour dans la page principale, pour activer le proxy, il suffit de cliquer sur le bouton *Proxy OFF*. Ce dernier passe à *Proxy ON* et vous indique qu'il est bien activé.

Si le proxy est mal configuré ou pas configuré du tout, un message d'avertissement viendra comme le montre la Figure ci-dessous

FIGURE 9 – Proxy mal configuré

2.5 Préférences

Revenons sur le panneau des préférences et expliquons un peu mieux chaque information qu'on peut y trouver.

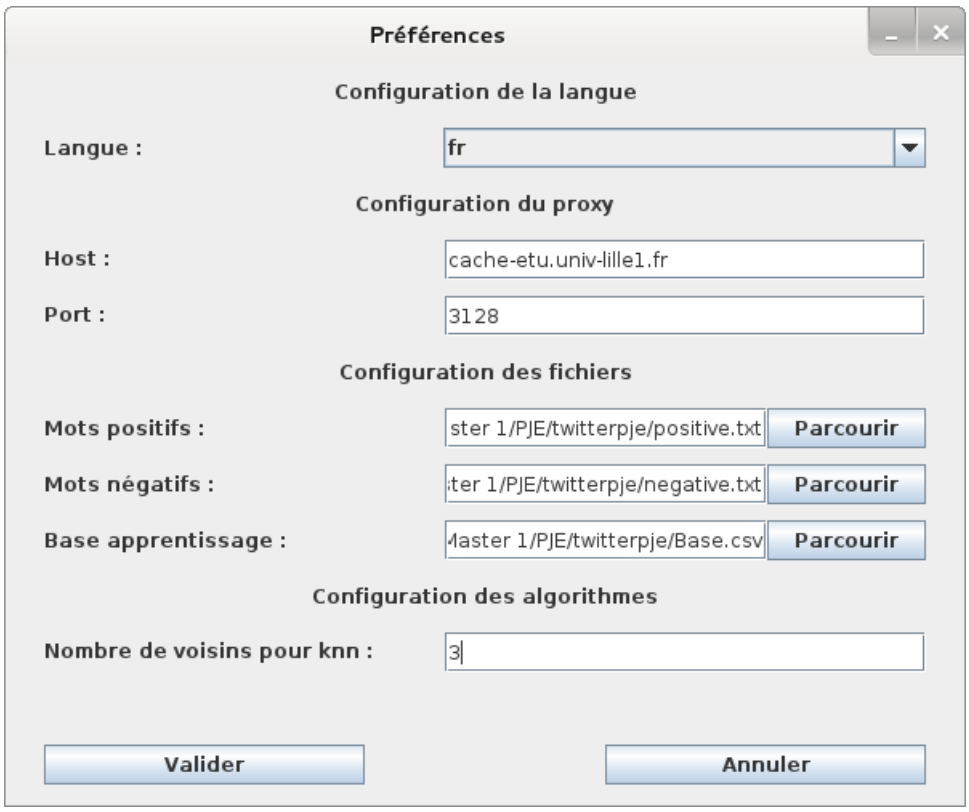


FIGURE 10 – Panneau des Préférences

La section *Configuration de la langue* permet de changer la langue des tweets qu'on souhaite rechercher (Par défaut, la langue est française).

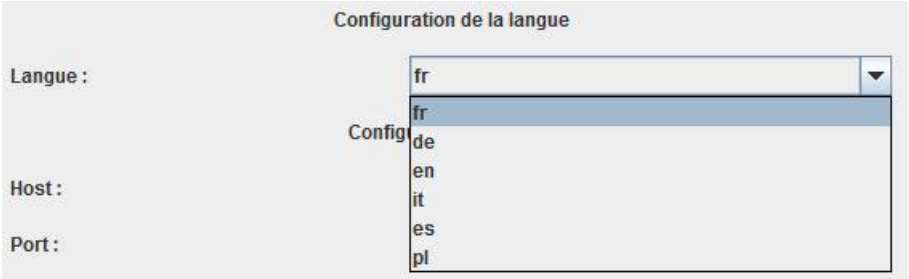


FIGURE 11 – Configuration de la langue

La section *Configuration des fichiers* permet d'affecter les dictionnaires de mots et la base d'apprentissage à utiliser pour les classifications.

Pour affecter un fichier, il faut cliquer sur le bouton *Parcourir* correspondant. Une fenêtre de sélection d'un fichier s'ouvre alors.

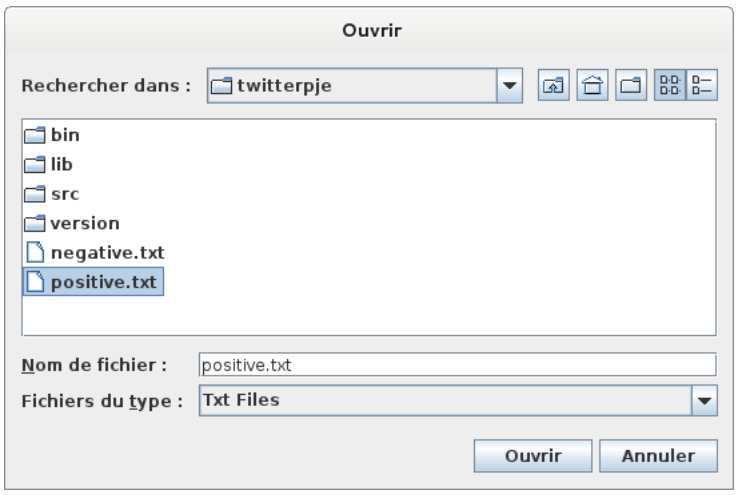


FIGURE 12 – Sélection du dictionnaire de mots positifs

Au clic sur le bouton *Ouvrir* , nous nous retrouvons dans le panneau des préférences avec le chemin absolu du fichier dans le champ correspondant (Voir Figure 10).

Le champ *Nombre de voisins pour KNN* sert comme son nom l'indique à déterminer le nombre de voisins pour l'algorithme de la classification KNN.

2.6 Classification

Maintenant que l'application est correctement configurée, nous pouvons passer au but principal de l'application, la classification.

Il existe plusieurs types de classifications et différentes manières de classifier, nous détaillerons chacune de ses classifications.

Notons que vous trouverez les différentes classifications dans l'onglet *Outils*

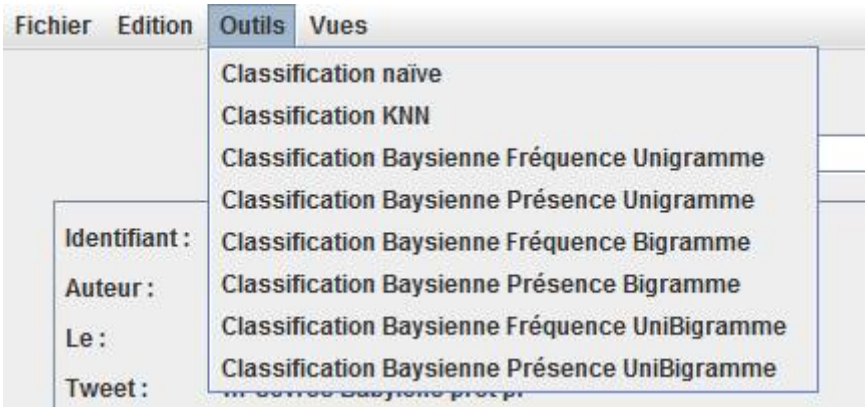


FIGURE 13 – L'onglet *Outils*

2.6.1 Manuelle

La première approche est de classifier manuellement chaque tweet dans la liste des tweets.

En effet, chaque tweet dans la liste est composé d'une ligne *Avis* suivi d'une liste déroulante avec *non annoté* comme valeur par défaut.

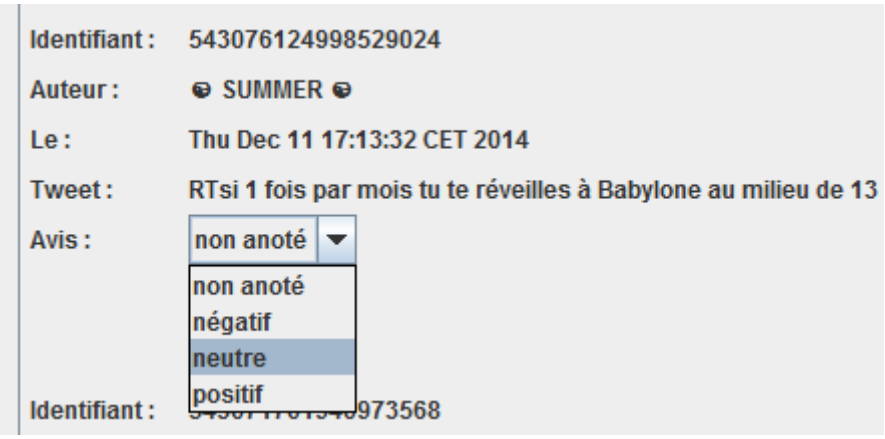


FIGURE 14 – Liste des sentiments

Modifier le sentiment d'un tweet met à jour en temps réel le graphique.

2.6.2 Par mots-clés

La seconde approche est de classifier d'un coup tous les tweets en fonction de dictionnaires de mots positifs et négatifs.

Attention à ne pas oublier d'importer les dictionnaires dans les préférences utilisateurs auquel cas, vous tomberez sur ce message :

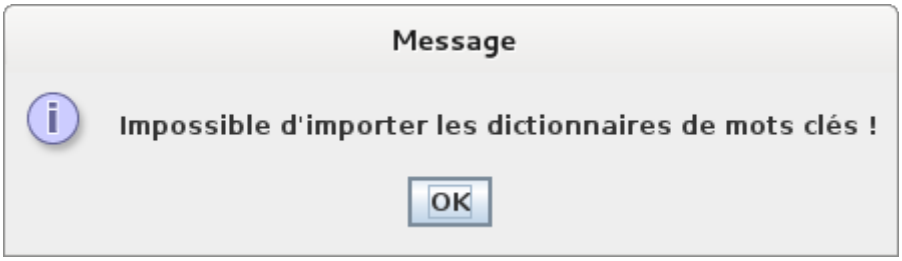


FIGURE 15 – Les dictionnaires ne sont pas importés

2.6.3 KNN

Le troisième type de classification est la classification KNN.

De même que pour la classification naïve, KNN a besoin d'une base d'apprentissage importée correctement ainsi qu'un nombre de voisins, auquel cas, vous tomberez sur ce message :

2.6.4 Bayésienne

La classification bayésienne est la dernière approche, elle n'a besoin que d'une base d'apprentissage pour fonctionner correctement, auquel cas, un autre type de message d'erreur vous sera renvoyé.

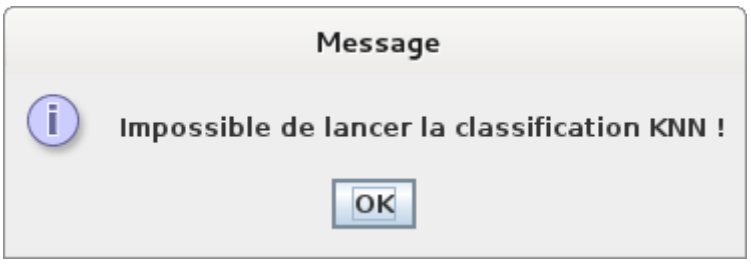


FIGURE 16 – La base d'apprentissage n'est pas importée correctement

2.7 Vues

L'application possède un dernier onglet *Vues* qui permet de permuter entre un histogramme ou un camember comme représentation graphique des sentiments à droite de l'application.

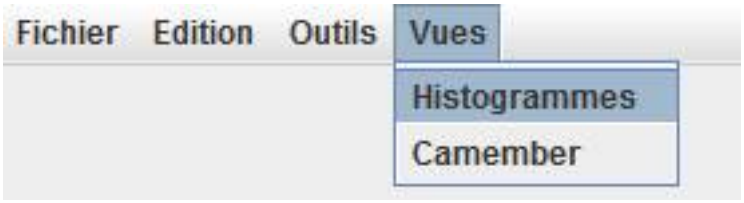


FIGURE 17 – L'onglet *Vues*

La Figure 3 nous montre l'application après une simple recherche sans aucune classification, voici les écrans après une classification bayésienne sous toutes les vues possibles :

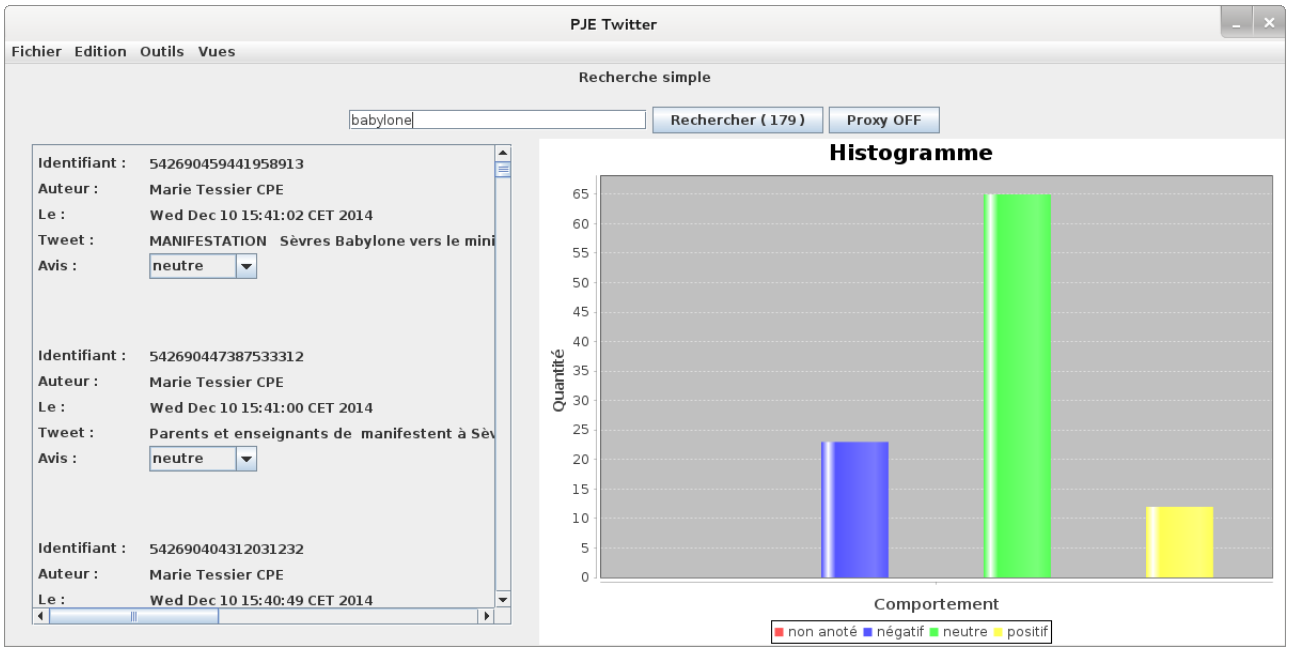


FIGURE 18 – Un histogramme

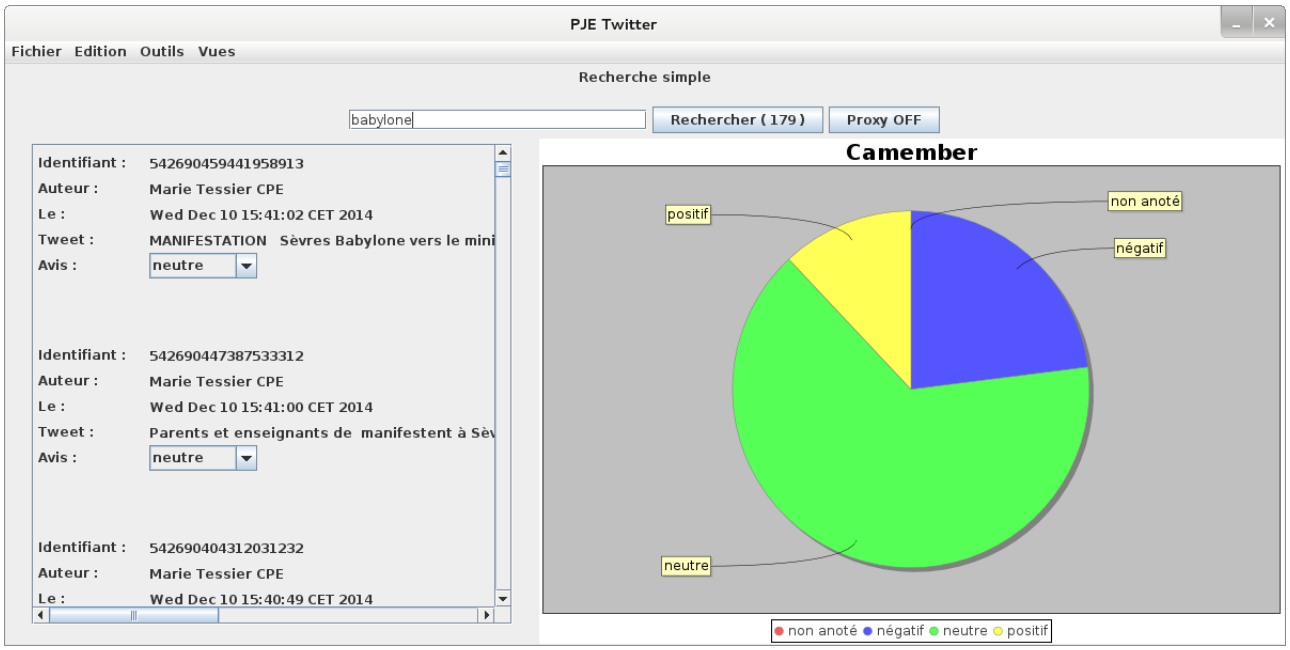


FIGURE 19 – Un Camember

3 Analyse et classifications

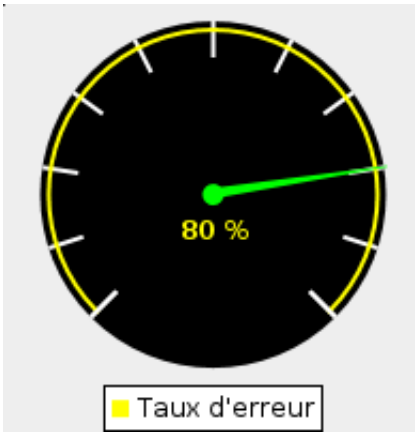
3.1 Base d'apprentissage

Avant de pouvoir effectuer toute classification, il est important d'avoir une base de tweets. Pour cela, il faut que l'utilisateur annote les tweets à la main. En effet, une classification manuelle restera toujours meilleur que des algorithmes.

Cette base sert donc de référence pour annoter les prochains tweets que l'on a récupéré via la recherche. Elle servira pour tous les algorithmes à l'exception de la classification par mots-clés.

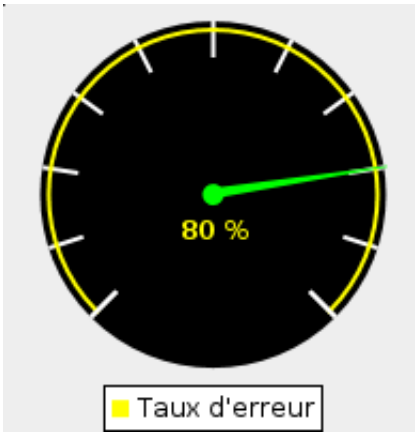
3.2 Classification par mots-clés

Topic : *hollande*



	Négatif	Positif	Neutre
Négatif	0	0	71
Positif	0	0	9
Neutre	0	0	20

Topic : *héros*

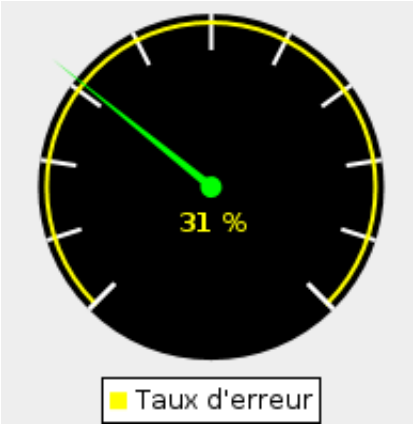


	Négatif	Positif	Neutre
Négatif	0	0	71
Positif	0	0	9
Neutre	0	0	20

3.3 Classification KNN

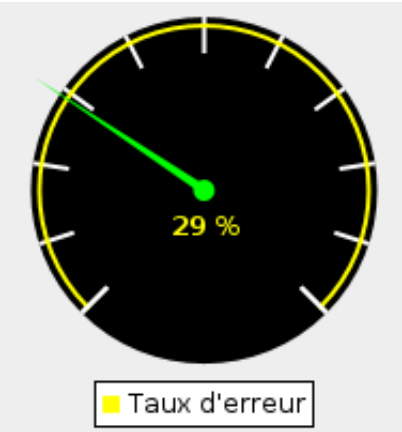
Topic : *hollande*

	Négatif	Positif	Neutre
Négatif	69	0	2
Positif	8	0	1
Neutre	20	0	0



Topic : *héros*

	Négatif	Positif	Neutre
Négatif	71	0	0
Positif	9	0	0
Neutre	20	0	0

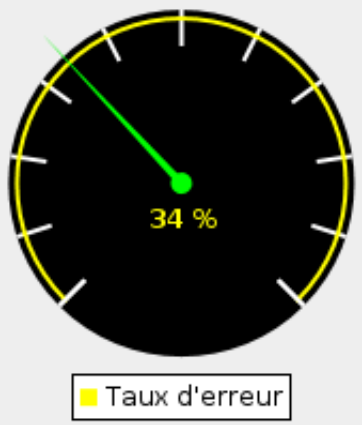


3.4 Classification bayésienne par fréquence

3.4.1 Unigrammes

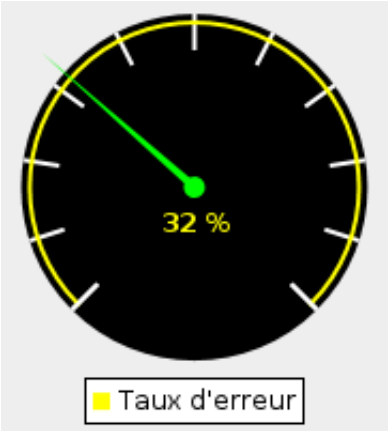
Topic : *hollande*

	Négatif	Positif	Neutre
Négatif	69	0	2
Positif	8	0	1
Neutre	20	0	0



Topic : *héros*

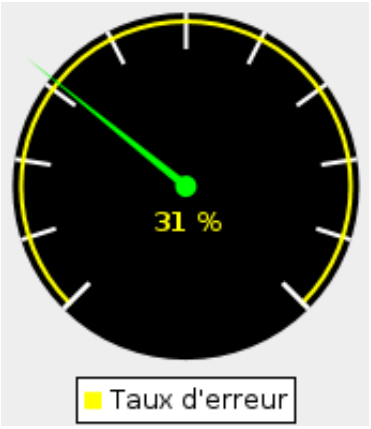
	Négatif	Positif	Neutre
Négatif	68	0	3
Positif	9	0	0
Neutre	20	0	0



3.4.2 Bigrammes

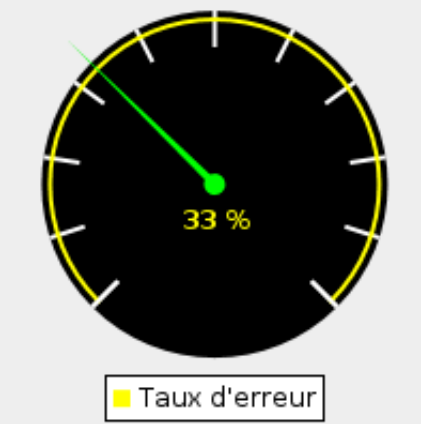
Topic : *hollande*

	Négatif	Positif	Neutre
Négatif	66	0	5
Positif	9	0	0
Neutre	20	0	0



Topic : *héros*

	Négatif	Positif	Neutre
Négatif	67	0	4
Positif	9	0	0
Neutre	20	0	0



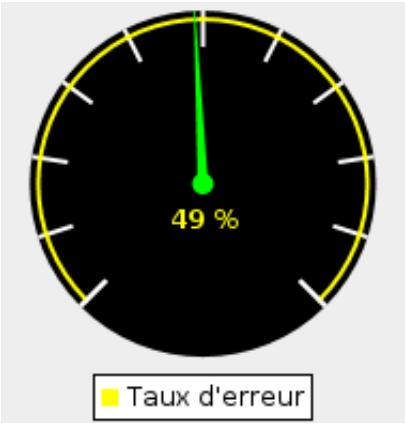
3.4.3 Unigrammes et bigrammes

Topic : *hollande*

	Négatif	Positif	Neutre
Négatif	54	1	16
Positif	7	0	2
Neutre	20	0	0

Topic : *héros*

	Négatif	Positif	Neutre
Négatif	49	1	21
Positif	6	0	3
Neutre	18	0	2

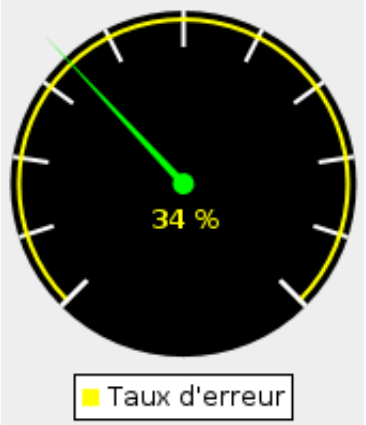


3.5 Classification bayésienne par présence

3.5.1 Unigrammes

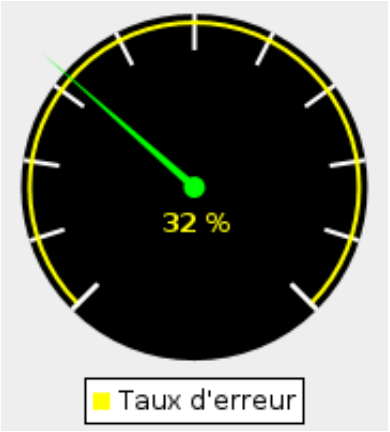
Topic : *hollande*

	Négatif	Positif	Neutre
Négatif	69	0	2
Positif	8	0	1
Neutre	20	0	0



Topic : *héros*

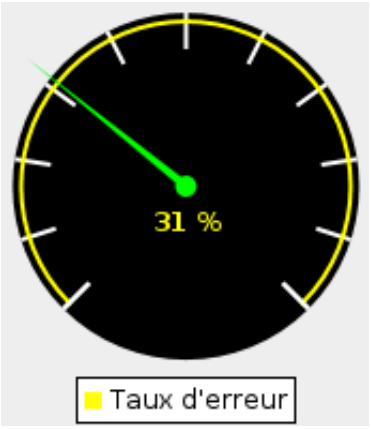
	Négatif	Positif	Neutre
Négatif	68	0	3
Positif	9	0	0
Neutre	20	0	0



3.5.2 Bigrammes

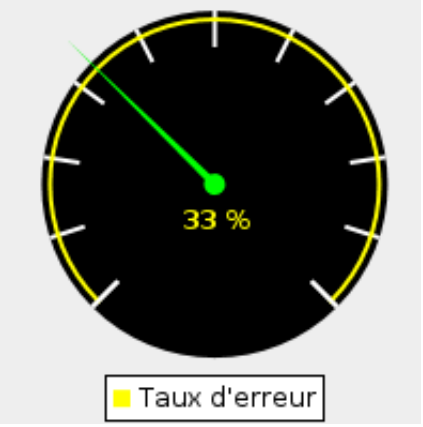
Topic : *hollande*

	Négatif	Positif	Neutre
Négatif	66	0	5
Positif	9	0	0
Neutre	20	0	0



Topic : *héros*

	Négatif	Positif	Neutre
Négatif	67	0	4
Positif	9	0	0
Neutre	20	0	0



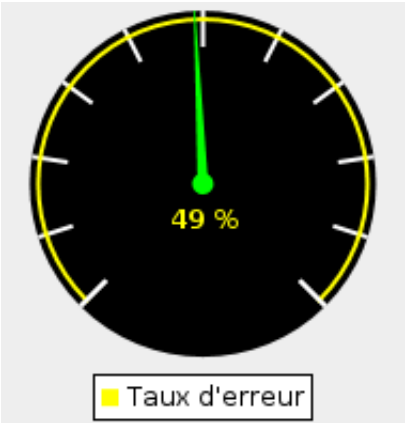
3.5.3 Unigrammes et bigrammes

Topic : *hollande*

	Négatif	Positif	Neutre
Négatif	54	1	16
Positif	7	0	2
Neutre	20	0	0

Topic : *héros*

	Négatif	Positif	Neutre
Négatif	49	1	21
Positif	6	0	3
Neutre	18	0	2



4 Comparatif global

	hollande	*héros*
Mots clés	20%	20%
KNN	69%	71%
Fréquence unigramme	66%	68%
Fréquence bigramme	69%	67%
Fréquence unigramme + bigramme	54%	51%
Présence unigramme	66%	68%
Présence bigramme	69%	67%
Présence unigramme + bigramme	54%	51%

Après analyse des résultats, on peut constater que les résultats sont assez différents en fonction des classifieurs. Cela dépend, bien entendu, de la base d'apprentissage mais aussi de la qualité des tweets. On remarque alors que les meilleurs classifieurs sont les bayésiens et KNN. Concernant la classification par mots-clés est très loin car en effet il faut renseigner tous les mots positifs et négatifs.Or, cela nécessite d'avoir un gros dictionnaire pour chaque sentiments. Donc ces classifieurs sont assez logiques au vu d'une analyse du sens des mots.

Conclusion

Dans notre projet, nous nous sommes intéressés au classements de certains algorithmes afin de trouver celui qui nous donne le meilleur résultat. Notre principal but principal est de créer un système d'analyse des sentiments automatique qui doit être adaptatif et indépendant de la langue.

Nous avons donc eu l'opportunité d'implémenter des algorithmes pour étiqueter les tweets disponibles sur Twitter. Cet étiquetage est juste une attribution d'un sentiment positif, négatif ou neutre. Pour certains algorithmes, nous devions disposer d'une base de référence de tweets déjà annotés à la main.

Lors de la réalisation de ce projet, nous avons donc développé de multiples compétences telles que la répartition du travail et l'autonomie.

En effet, nous avons dû nous organiser avec rigueur pour effectuer nos tâches respectives. Nous avons également appliquer une méthode Agile : Scrum. Pour nous, nos *sprints* avaient une durée d'une semaine et nous avions chacun des taches à faire. Grâce à cette méthode, nous avons géré parfaitement notre temps et avons pu d'être le plus efficace possible. Sans cette méthode, il nous aurait été difficile de mener à bien ce projet avec ses contraintes et difficultés.

Pour conclure, on peut affirmer que ce projet nous a été très instructif et a amélioré nos connaissances.