ANALYSE DES TWEETS LIÉS AUX APPLICATIONS DE SURVEILLANCE EN UTILISANT TALEND

https://github.com/savacano28/analysis-twitter

1. Description du projet

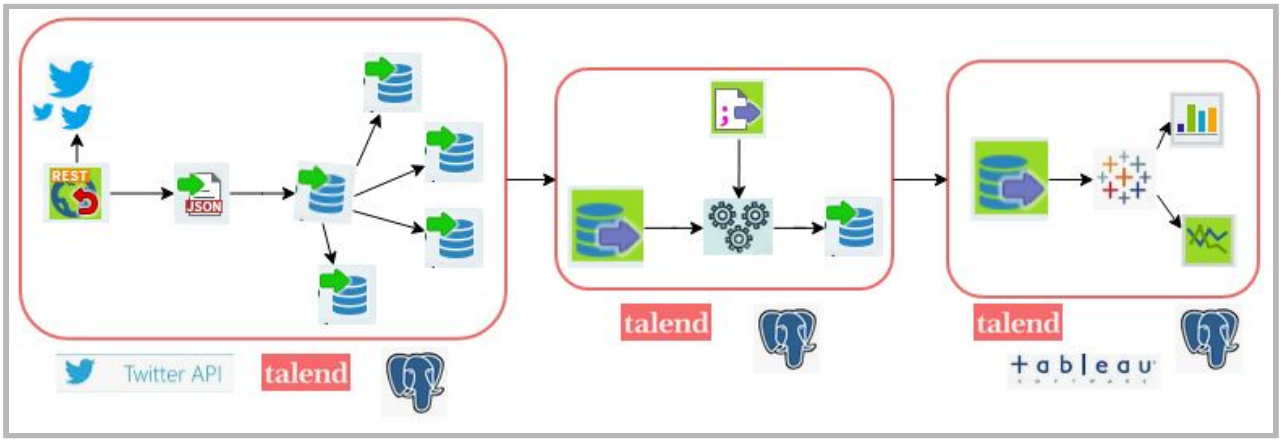
Dans ce projet, nous avons **analysé les tweets publiées depuis le 15 novembre 2020 liés au développement et à l'utilisation d'applications mobiles de surveillance du COVID dans 3 pays : France, Angleterre et Allemagne**. Pour atteindre cet objectif, nous avons utilisé Talend pour réaliser les étapes de récupération (connexion à l'API REST Twitter), data cleaning, stockage (connexion et crud á la bdd), traitement et modélisation d'entrepôt de données. Les résultats obtenus ont été visualisés sur l'application tableau.

1.1 Source de Données

La source de données correspond aux opinions exprimées sur le réseau social **Twitter** liées aux applications mobiles de tracade de COVID. Nous avons considéré les tweets publiés entre le 15 oct 2020 et le 09 déc 2020 avec les hashtags *#tousAntiCovid* (France app), *#NHSCOVID2019* (Anglettere app) *et #coronaWarnApp* (Allemagne app).

1.2 Technologies utilisés

Postgres 12.0 **Talend 7.0** **Twitter API** **Tableau**

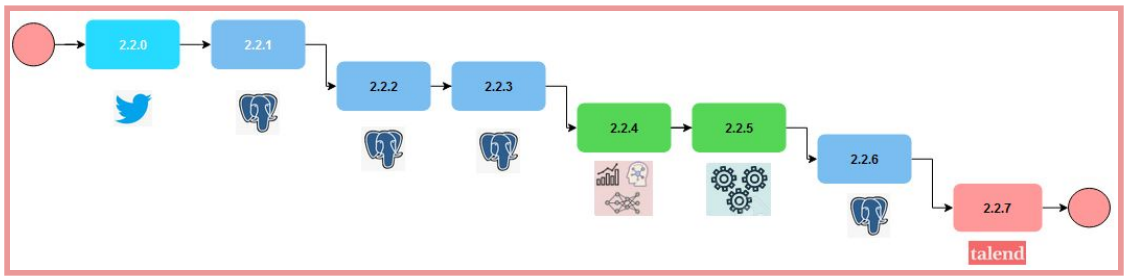


1.3 Architecture proposée

Architecture

2. Mis en place du projet sur Talend

Workflow du projet construit sur talend. Chaque numéro correspond à une étape.



workflow du projet

2.1 Configurations utilisées

Nous avons défini plusieurs variables de contexte tels que :

- **port, server, login et pwd** d'accès à la bdd postgres
- **path du répertoire de travail** où seront sauvegardés les fichiers générés par talend

	Name	Type	Comment	dev Value
1	connection_postgres_Server	String		
2	connection_postgres_Login	String		
3	connection_postgres_Port	String		
4	connection_postgres_Passwo	String	admin	
5	connection_postgres_Databa	String	talend	
6	connection_postgres_Schema	String		
7	connection_postgres_Additio	String		

	Name	Type	Comment	dev Value
1	path_files_global	String		?

Configuration de variables de contexte

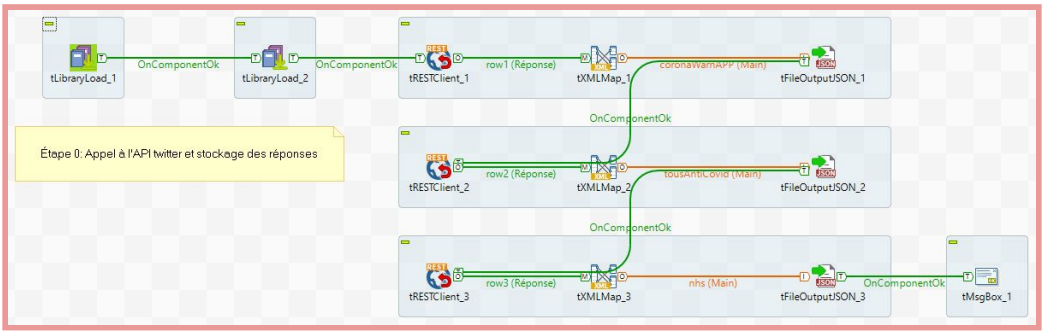
2.2 Étapes - jobs

2.2.0 Connexion api twitter

L'api de twitter -version gratuite- permet la récupération des tweets des 7 derniers jours et de 200 tweets (limitation de twitter). Le méthode d'authentification est en utilisant un token d'accès qui s'obtient une fois que l'on s' inscrit comme développer twitter. Les tweets qui nous avons récupéré sont ceux qui contenaient comme hashtag les nom des app COVID.[1]

Parametres	Value
Authentication	OAuth2 Bearer
Rq #coronaWarnApp	"https://api.twitter.com/1.1/search/tweets.json?q=%**23coronaWarnApp**&result_type=recent&count=1000&lang=en"
Rq #tousAntiCovid	"https://api.twitter.com/1.1/search/tweets.json?q=%**23TousAntiCovid**&result_type=recent&count=1000&lang=fr"
Rq# NHSCOVID19	"https://api.twitter.com/1.1/search/tweets.json?q=%**23NHSCOVID19app**&result_type=recent&count=1000&lang=en"

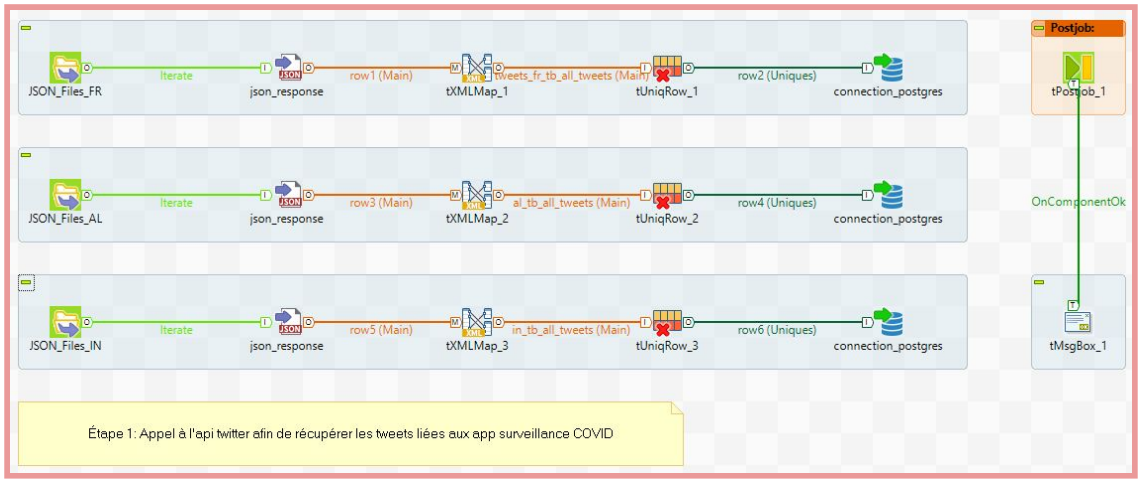
Les composants utilisés ont été **tRestClient** (connexion à api rest twitter), **tmap** (pour récupérer juste les champs qui nous intéressent comment localisation, date, user, text…) et **tFileOutputJson**.



S0. Récupération des tweets liés aux app covid

2.2.1 Téléchargement des tweets dedans de la base de données

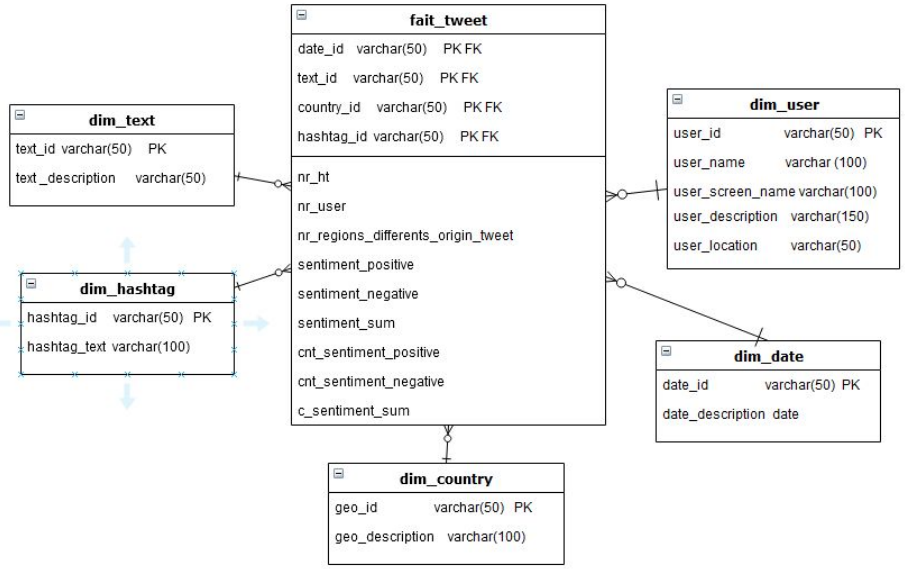
Après d'avoir sauvegardé les réponses de l'api twitter en fichiers json pendant 25 jours, nous avons centralisé cette information dans la bdd *talend* dans la table *all_tweets,* pour réaliser cette action nous avons utilisé les composants suivants : **tFileList** (afin de boucler dans le répertoire où nous avons sauvegardé les fichiers json), **tFileInputJson, tMap** (mappear et transformer les attributes qui nous intéressant), **tUniqueRow** (nous avons trouvé que l'enregistrement de chaque jour des tweets pouvait avoir des tweets répétés puisque sont les 200 derniers, et si dans 2 jours personne publié des tweets donc nous allons recevoir 2 fois les mêmes tweets)



S1 Téléchargement des tweets dedans de la base de données

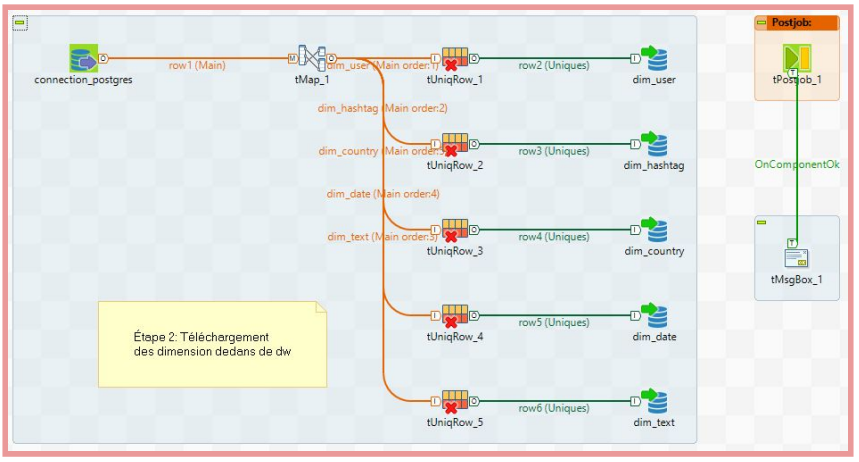
2.2.2 Téléchargement des dimension dans le dw

Dans cette étape nous avons construit le DW à partir des données stockées dans la table *all_tweets*. Nous avons identifié 5 dimensions : dim_user, dim_text, dim_country, dim_hashtag, dim_date.



Modele dw twitter

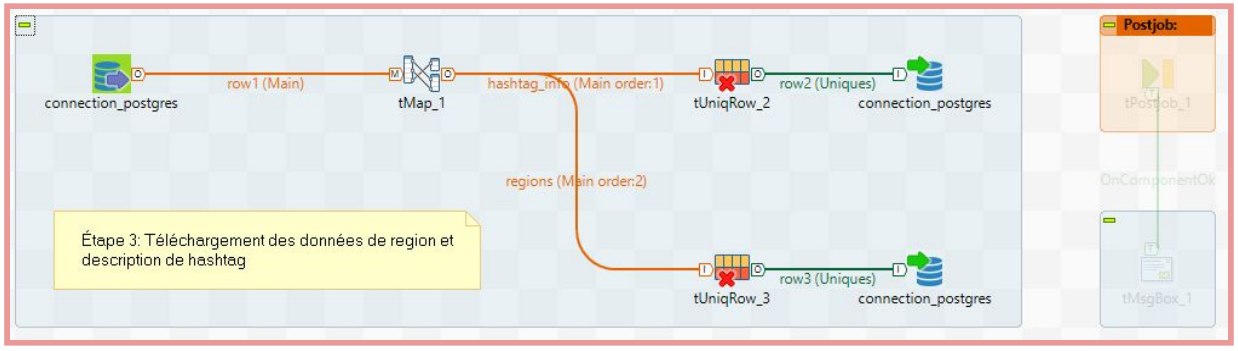
Le chargement des données dans les dimensions à été réalisé en utilisant les composants: **tDbInput** (pour lire depuis la table *all_tweets*), **tMap, tUniqueRow, tDbOuput** (pour écrire dans les tables des dimensions).



S2 Téléchargement des dimension dans le dw

2.2.3 Téléchargement des autres données dans dw

Dans cette étape nous avons souhaité conserver des informations plus détaillées sur les hashtags et les régions d'origine des tweets. Ici, nous avons procédé de la même manière que l'étape précédent, la différence est dans le tMap où nous avons choisi les attributs qui nous intéressent.



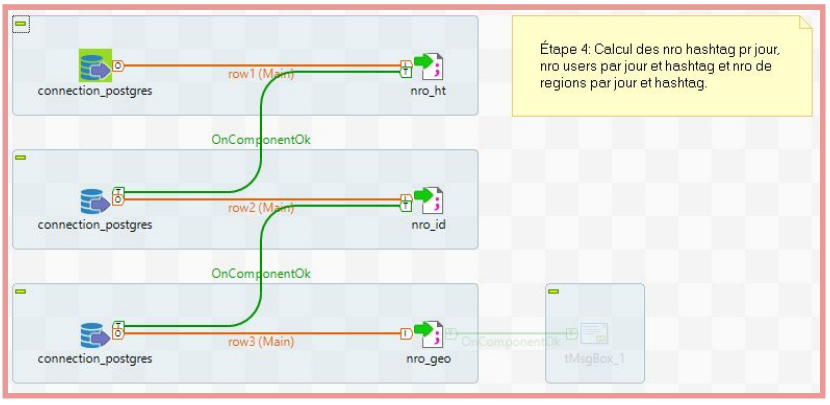
S3 Téléchargement des données dans le dw

2.2.4 Calcul des mesures

Après avoir chargé les données dans les tables de dimensions, nous avons calculé les différentes mesures. Dans le tableau ci-dessous vous pouvez visualiser la requête effectuée pour chaque mesure.

Mesure	Description	sql
no_ht	Nombre de hashtags par date et pays	"select to_char(d.date,'dd-MM-yyyy') as date,d.hashtag as hashtag,d.country as country, count(distinct d.id) as no_ht from all_tweets d group by d.date, d.hashtag,d.country order by d.date asc;"
no_regions	Nombre de régions	"select to_char(d.date,'dd-MM-yyyy') as date,d.hashtag as hashtag,d.country as country,count(distinct d.geo) as no_regions from all_tweets d group by d.date, d.hashtag,d.country order by d.date asc;"
no_users	Nombre d' usagers	"select to_char(d.date,'dd-MM-yyyy') as date,d.hashtag as hashtag,d.country as country,count(distinct d.user_id) as no_users from all_tweets d group by d.date, d.hashtag,d.country order by d.date asc;"

Les composants utilisés sont : **tDbInput** (pour faire les requêtes respectives à chacune des mesures), **tFileDelimitedOutput**.

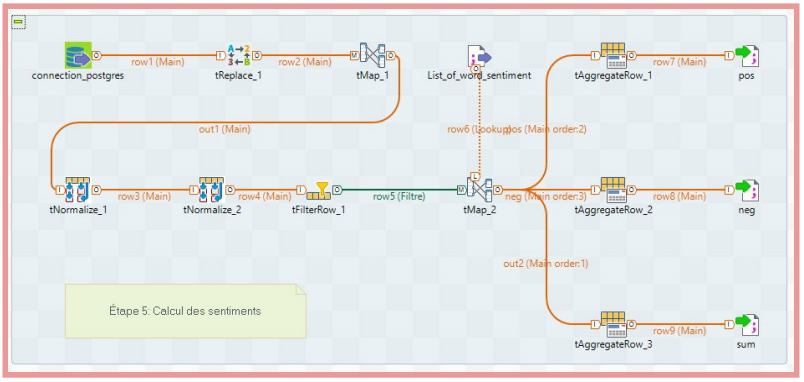


S4 Calcul des mesures

2.2.5 Calcul de sentiments

Dans cette étape, nous avons calculé les sentiments liés aux tweets avec hashtags #tousAntiCovid, #coronaWarnApp et #NHScovid19. Pour débuter nous avons récupérer l'attribut "text" de chaque tweet et nous avons effectué des traitements sur chaque tweet : supprimer les caractères étranges, split sur les mots composant ce tweet, supprimer les mots nulls, après nous avons comparé chaque mot avec notre dictionnaire de sentiments ce qui nous as permis de donner un poid à chaque mot, ensuite nous avons regroupé chaque mot par l'id de son tweet et, pour finaliser, nous avons additionné leur poids.

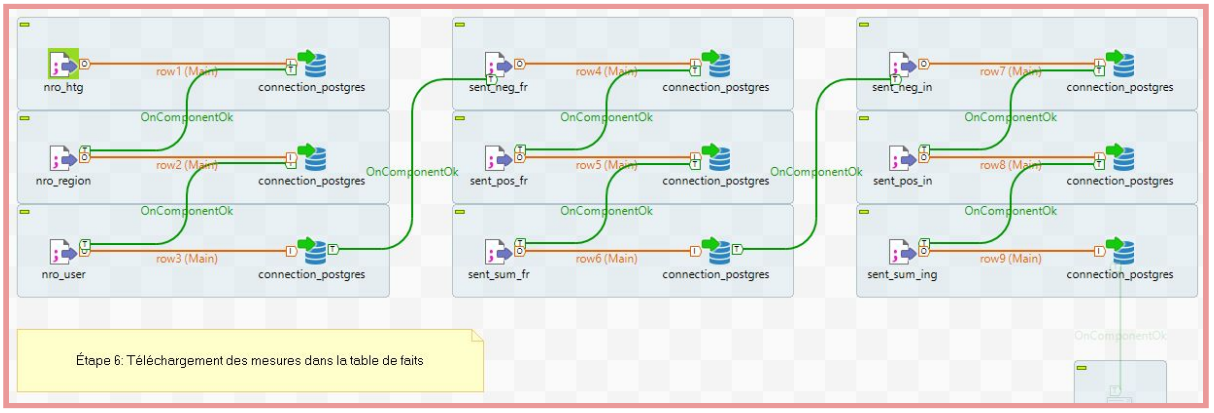
Les composant utilisés sont : **tReplace** (supprimer des character étranges), **tNormalize** (pour splitter chaque tweet sur les mots qui le composent), **tFilterRow** (filtrer des lignes nulls), **tmap** (pour faire le join entre le mot et notre dictionnaire de sentiments et récupérer son poid, par exemple poid > 0 est un sentiment de soutien et poid <0 sentiment de rejet, **tAggregateRow** (calcul le poid global du tweet à partir de poid de ses mots en groupant les mots selon le tweet auquel ils appartiennent), **tFileDelimitedOutput**.



S5 Calcul des sentiments

2.2.6 Téléchargement des mesures dans le dw

Finalement, nous avons alimenté tous les faits dans la table de fait. Les composant utilisés sont : **tFileDelimitedInput**, **tDbOutput** (pour écrire les faits dans la table *fait_tweet.*



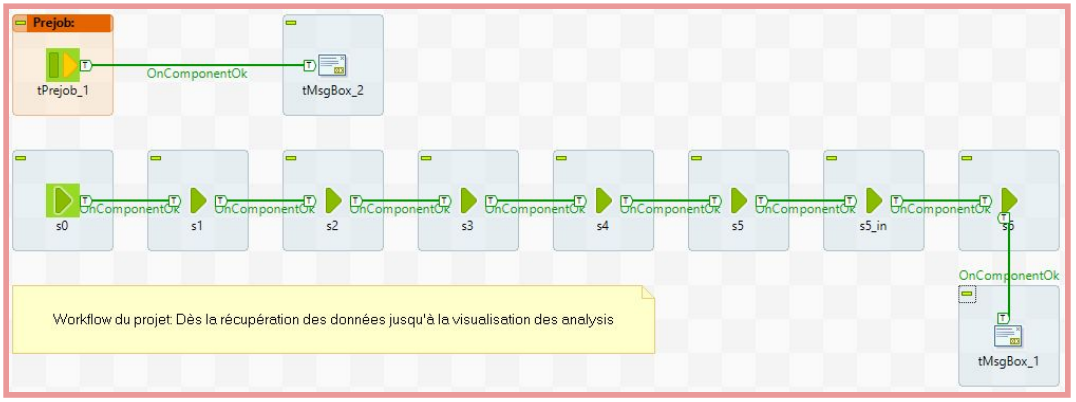
S6 Téléchargement des mesures dans la table de fait

2.2.7 Workflow final pour le projet in talend

Dans ce dernier job, nous avons créé le flux de travail complet pour le projet :

jobS0 -> jobS1 -> jobS2 -> jobS3 -> jobS4-> jobS5 -> jobS6

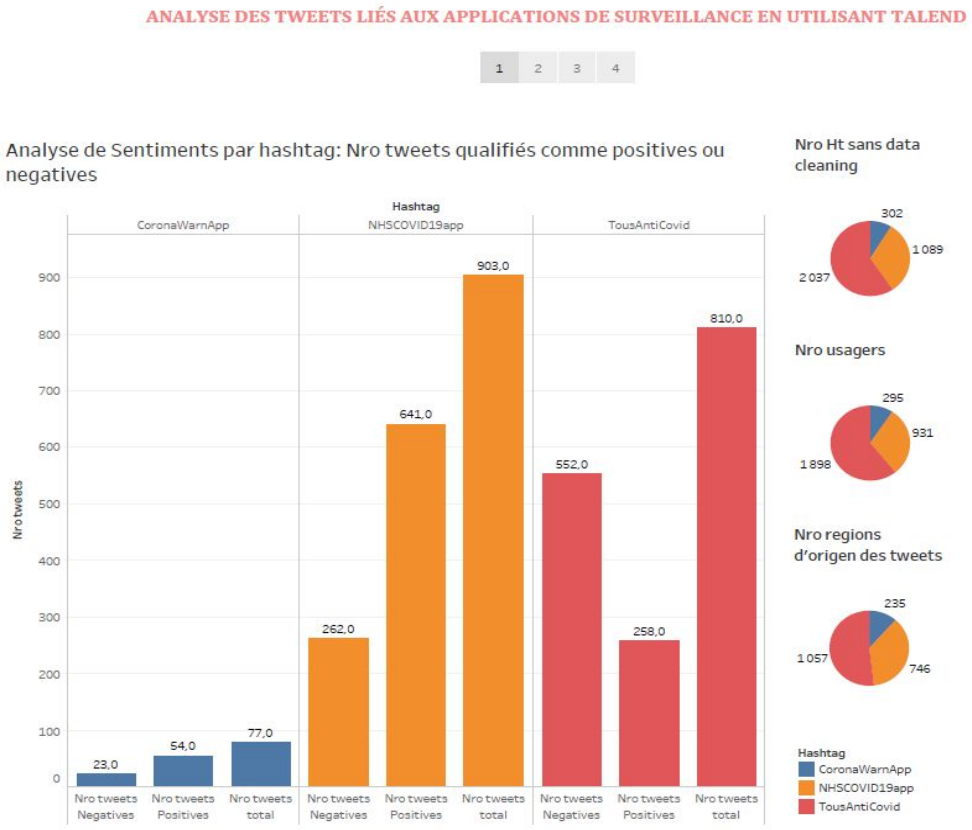
Il faut mentionner que le **contexte** est complètement transmis depuis les jobs parents vers les jobs enfants, donc les **paramètres de contexte pour la connexion à la bdd talend en postgres et le path du répertoire de travail** pour sauvegarder les fichier produit sont le **même dans chaque job**.



S7 WorkFlow final

2.2.8 Visualisation sur Tableau - Dashboard

Dans Tableau nous avons fait une connexion vers notre bdd talend (postgresql) et nous avons construit un dashboard pour afficher les résultats de notre projet d'analyse de tweets liés aux app COVID.



3. Livrables du projet

Projet talend (zip)	~downloads/analysis-twitter/
Executables (sh)	~downloads/analysis-twitter/documentations/executables/analysis_twitter_dw_pj/workflow_final/workflow_final_run.sh
Guide d'exécution (pdf)	~downloads/analysis-twitter/documentations/ressources/Lisez-moi.pdf
Documentation du projet (pdf)	~downloads/analysis-twitter/documentations/ressources/Projet_Talend.pdf
Dashboard tableau (pdf)	~downloads/analysis-twitter/documentations/ressources/Histoire_analyse_twitter.pdf
Vidéo explicative projet en talend (mp)	~downloads/analysis-twitter/documentations/ressources/analyse_twitter.mp4 ~downloads/analysis-twitter/documentations/ressources/lancer_executable_talend.mp4

References

[1] https://developer.twitter.com/en/docs/twitter-api (25-10-20)
[3] https://www.datalytyx.com/twitter-sentiment-analysis-using-talend/(22-10-20)
[4]https://www.la-croix.com/Monde/Europe/INFOGRAPHIE-Coronavirus-tour-dEurope-applications-tracage-contacts-2020-06-03-1201097309 (25-10-20)