Traitement de l'information

Ophélie Fraisier

2018 – 2019

Vous allez réaliser une étude approfondie d'un jeu de données de tweets.
— Équipes de 3 ou 4
— Un hashtag à étudier par équipe
`#starwars`, `#marvel`, `#trump`, `#macron`, `#toulouse`, `#paris`, `#android`, `#ios`, ...
— Jeu de données d'au moins 30 000 tweets.
— **Présentations orales et rendu le 30/11/18**

1 Documents à rendre

— Un rapport présentant vos résultats (en **pdf**).
— Le support de votre présentation.
— Une archive contenant les scripts et fichiers utilisés.

1.1 Indications sur le rapport

— Une organisation claire est attendue, avec plan, introduction et conclusion. Vous êtes libres de définir les sections qui vous semblent les plus pertinentes.
— Utilisez des graphiques pour illustrer vos découvertes.
— Pour chaque analyse, indiquez en quoi elle vous semble pertinente et les informations que vous souhaitez en tirer.
— Annexes possibles : extraits de scripts réalisés expliqués, quelques exemples d'articles de presse présents dans le jeu de données, ...

Remarque importante : si une analyse vous donne des résultats peu exploitables, il s'agit d'un résultat néanmoins. Incluez-la dans votre rapport en indiquant pourquoi, à votre avis, elle a donné de mauvais résultats.

1.2 Indications sur la présentation

— 10 à 15 minutes par groupe + 5 minutes de questions / remarques
— **Tous** les membres du groupe doivent parler
— Présentation du contexte et des résultats les plus pertinents sur votre jeu de données

2 Quelles analyses ?

2.1 Analyses obligatoires

Les analyses qui figurent ci-dessous doivent obligatoirement apparaître dans votre rapport.

- Aperçu général de l'étude
  - Hashtag étudié
  - Raisons du choix du hashtag
  - Taille du jeu de données
  - Date et durée de la collecte
- Étude sur les tweets
  - Hashtags présents (`status["entities"]["hashtags"]`)
  - Proportion de retweets
  - Tweets les plus retweetés
  - Répartition par jour et / ou par heure (utilisez la / les granularité(s) les plus pertinentes pour votre étude, voir Figure 1) (`status["created_at"]`)
  - Corrélation entre le nombre de tweets et le jour ou l'heure (voir cours sur la régression linéaire)
  - Réseau de co-hashtags, voir Figure 3 et Figure 4)
- Étude des profils
  - Nombre de profils dans le jeu de données
  - Les plus présents dans le jeu de données (voir Tableau 1)
  - Les plus influents (préciser quelles mesures vous considérez pour calculer l'influence) (`status["user"]["followers_count"]`, `status["user"]["friends_count"]`, `status["user"]["statuses_count"]`, `status["user"]["listed_count"]`)
- Étude des médias externes
  - Urls les plus partagées (voir Tableau 2) (`status["entities"]["urls"]`)

2.2 Analyses supplémentaires au choix

En plus des analyses obligatoires indiquées ci-dessus, vous devrez réaliser 5 analyses de votre choix.

Vous trouverez ci-dessous quelques exemples pour vous inspirer, mais vous êtes libres d'inclure les analyses de votre choix, du moment que vous pouvez expliquer en quoi elles aident à explorer le jeu de données.

- Répartition du nombre de tweets par profil (moyenne, médiane, écart-type, boxplot, ...)
- Nuage des mots les plus fréquents (voir Figure 2)
- Profils les plus mentionnés (possibilité de faire un réseau de mention)
- Lieux présents (`status["user"]["location"]`)
- Médias présents (Le Monde, Le Figaro, Le Gorafi, ...)
- Récupération et analyse des articles de presse cités
- Catégories des articles (politique, faits divers, ...)
- Nuage de mots des articles les plus partagés (titres ou contenus complets)
- Classification des sentiments
- Classification des thèmes

Pour ces 2 derniers points, vous pouvez par exemple utilisez uClassify qui proposent de nombreux classifieurs : `https://uclassify.com/browse`

- Sentiment : `https://uclassify.com/browse/uclassify/sentiment`
- Thèmes : `https://uclassify.com/browse/uclassify/topics?input=Text`

Documentation des APIs : `https://uclassify.com/docs/apidifferences`

3 Outils et ressources en ligne

3.1 Twitter

— Documentation de l'API : `https://dev.twitter.com/rest/reference`
— Documentation de la fonction search/tweets (incluant la structure JSON d'un tweet) : `https://dev.twitter.com/rest/reference/get/search/tweets`
— Les opérateurs utilisables dans la requête : `https://dev.twitter.com/rest/public/search`

3.2 Nuages de mots

— WordItOut : `https://worditout.com`
— WordClouds : `http://www.wordclouds.com`
— Jason Davies' Word Cloud Generator : `https://www.jasondavies.com/wordcloud/`
— Tagul : `https://tagul.com`

Comme vous l'avez sûrement remarqué lors du TP précédent, les nuages de mots peuvent être bruités par les mots les plus courants : la, les, du, une, est, ... Ces mots ne présentant pas d'intérêts pour l'interprétation des nuages, ils sont appelés *mots vides* ou *stop words*, et il est préférable de les supprimer lors des analyses textuelles.

Vous trouverez à l'adresse suivante des listes de mots vides pour plusieurs langues si vous souhaitez les utiliser lors du projet : `https://code.google.com/archive/p/stop-words/`

3.3 Gephi

— Documentation officielle : `https://gephi.org/users/`
— Tutoriel basique : `bibliotheques.wordpress.com/2014/07/23/gephi-premiere-utilisation-spatialisation`

4 Scripts fournis

Les scripts décrits ci-dessous sont fournis pour vous aider à analyser votre jeu de données. ***Modifiez-les à votre convenance afin de personnaliser et approfondir votre étude de cas.***

4.1 Récupération du jeu de données

Le script `get_dataset.py` est fourni pour vous aider à collecter vos jeux de données. Il s'agit d'une version minimale que vous pouvez modifier comme vous le souhaitez, la seule condition étant que votre jeu de données final contienne au moins 30 000 tweets. Il se lance à l'aide de la commande suivante :

```
python3 get_dataset.py "<keyword>" <size>
```

avec `<keyword>` le mot-clé permettant de filtrer les tweets et `<size>` la taille du dataset souhaité [1].

Si votre mot-clé est un hashtag, n'oubliez par **le #** en début de mot.

Le dataset sera stocké dans le fichier `dataset_<keyword>_<size>.json` placé dans le même répertoire que le script.

Pour plus d'informations, consultez la documentation de la fonction search tweets de l'API Twitter : `https://dev.twitter.com/rest/reference/get/search/tweets`

Par exemple, pour récupérer un jeu de données de 5 000 tweets contenant #Toulouse, la commande est la suivante : `python3 get_dataset.py "#Toulouse" 5000`

Remarque : sur Twitter, la casse n'est pas prise en compte dans les mots-clés. Vous pouvez donc indifféremment chercher pour #Toulouse ou #toulouse.

1. Dans les faits le nombre de tweets récupérés sera toujours un multiple de 100. Si vous indiquez donc une taille de 550 tweets, votre jeu de données en contiendra 600 au final.

Nom d'utilisateur	Nom complet	Nombre de tweets postés
stevenbrower475	Cara Green	151
elizabe99742652	Bella Bailey	146
lunglesbee977	Jay Anderson	134
deleoly85	Alina Young	134
holmir385	Becky Taylor	133
frebar86	Lexi Flores	132
crrzy69	Kitty Jackson	131
florencioostro1	Janet Hughes	128
tahrirlive	Tahrir Live News	124

TABLE 1 – Utilisateurs les plus actifs sur le sujet des attentats de Berlin le 20/12/2016

4.2 Répartition temporelle

Le script `dataset_treatments.py` contient une fonction permettant de voir la répartition des tweets dans le temps. Vous pouvez l'appelez à l'aide de la commande suivante :

```
python3 dataset_treatments.py <dataset_file> -d <option>
```

avec `<option>` dans la liste suivante :
- `days` : regroupe par jour de la semaine (1=lundi, 2=mardi, etc),
- `hours` : regroupe par heure de la journée (de 0 à 23),
- `dates_days` : regroupe par jour du calendrier,
- `dates_minutes` : regroupe par jour du calendrier et minute.

Il s'agit bien évidemment là de 4 exemples de découpages temporels courants, mais vous pouvez exploiter les découpages temporels de votre choix. Ce script affiche ses résultats au format CSV dans le terminal mais vous pouvez utiliser l'opérateur `>` pour rediriger la sortie standard dans un fichier.

```
python3 dataset_treatments.py <dataset_file> -d <option> > output_file.csv
```

Quelques exemples d'utilisation sont présentés dans la Figure 5.

Documentation du type date en Python3 : `https://docs.python.org/3.6/library/datetime.html#datetime-objects`

4.3 Extraction des co-hashtags

Le script `dataset_treatments.py` contient également une fonction permettant d'extraire les co-hashtags présents dans les tweets. Vous pouvez l'appelez à l'aide de la commande suivante :

```
python3 dataset_treatments.py <dataset_file> -# [list of hashtags to ignore]
```

Dès qu'il rencontre 2 co-hashtags dans un tweet, ce script affiche dans le terminal «hashtag1,hashtag2». Là encore vous pouvez utilisez l'opérateur `>` pour stocker ces résultats dans un fichier. Le CSV ainsi produit peut-être ouvert dans Gephi afin de visualiser le graphe des co-hashtags.

Remarque : pensez à ignorer le hashtag avec lequel vous avez construit votre jeu de données. Il n'apportera pas d'information pertinente étant donné qu'il est, par construction, présent dans tous les tweets de votre jeu de données.

Par exemple : `python3 dataset_treatments.py dataset_#Toulouse_5000.json -# "#toulouse" "#bordeaux" "#montpellier" > toulouse_cohashtags.csv`

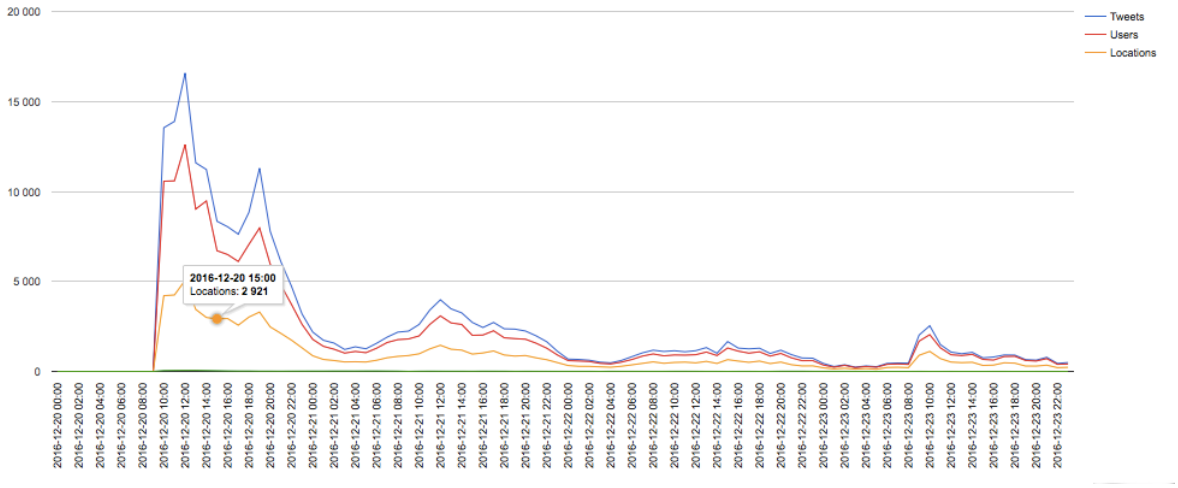

FIGURE 1 – Évolution du nombre de tweets, d'utilisateurs, et de tweets géolocalisés portant sur l'attentat de Berlin dans le temps

FIGURE 2 – Nuages des mots les plus utilisés dans les tweets portant sur l'attentat de Berlin du 20 au 23/12/2016

Url	Fréquence
https://www.rt.com/news/370969-berlin-police-detained-wrong-man/	1053
https://www.rt.com/news/370837-berlin-truck-christmas-market/	896
http://www.philstockworld.com/2016/12/20/view-single-comment-6396191/	1023
http://www.bbc.co.uk/news/live/world-europe-38370364	838
http://www.tiredearth.com/	755

TABLE 2 – Urls les plus partagées sur le sujet des attentats de Berlin le 20/12/2016

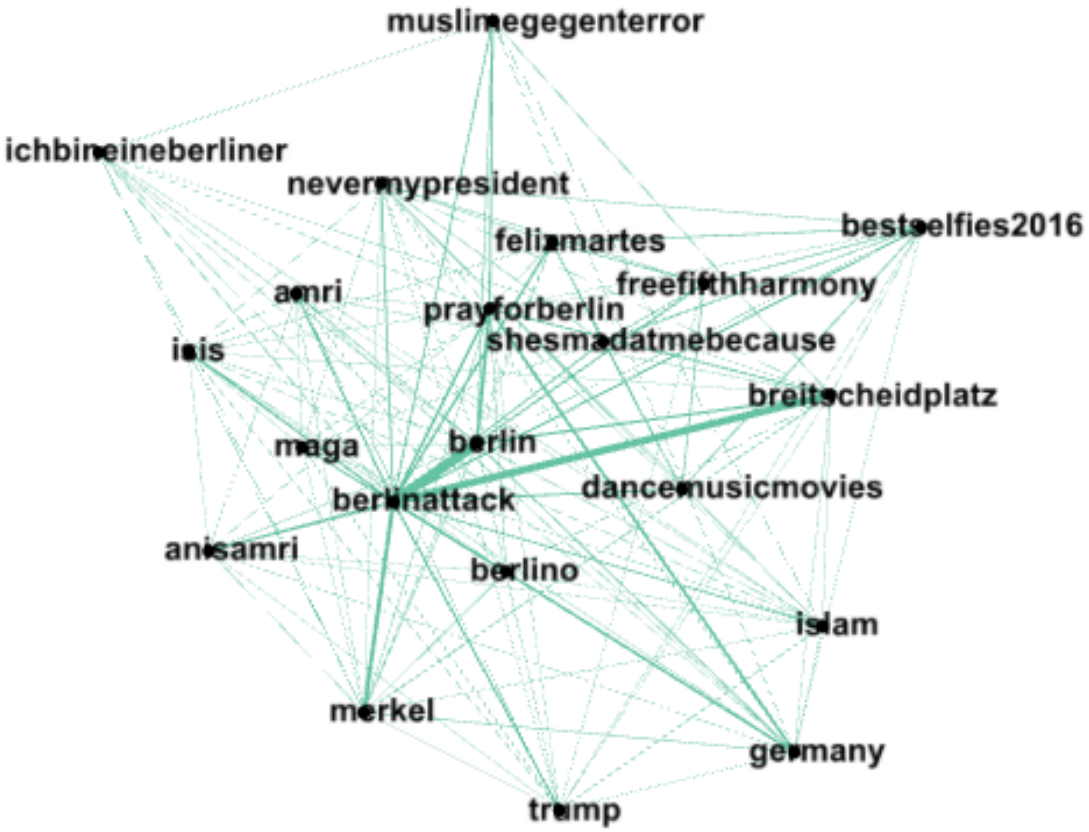

FIGURE 3 – Réseau de co-hashtags présents dans les tweets portant sur l'attentat de Berlin du 20 au 23/12/2016

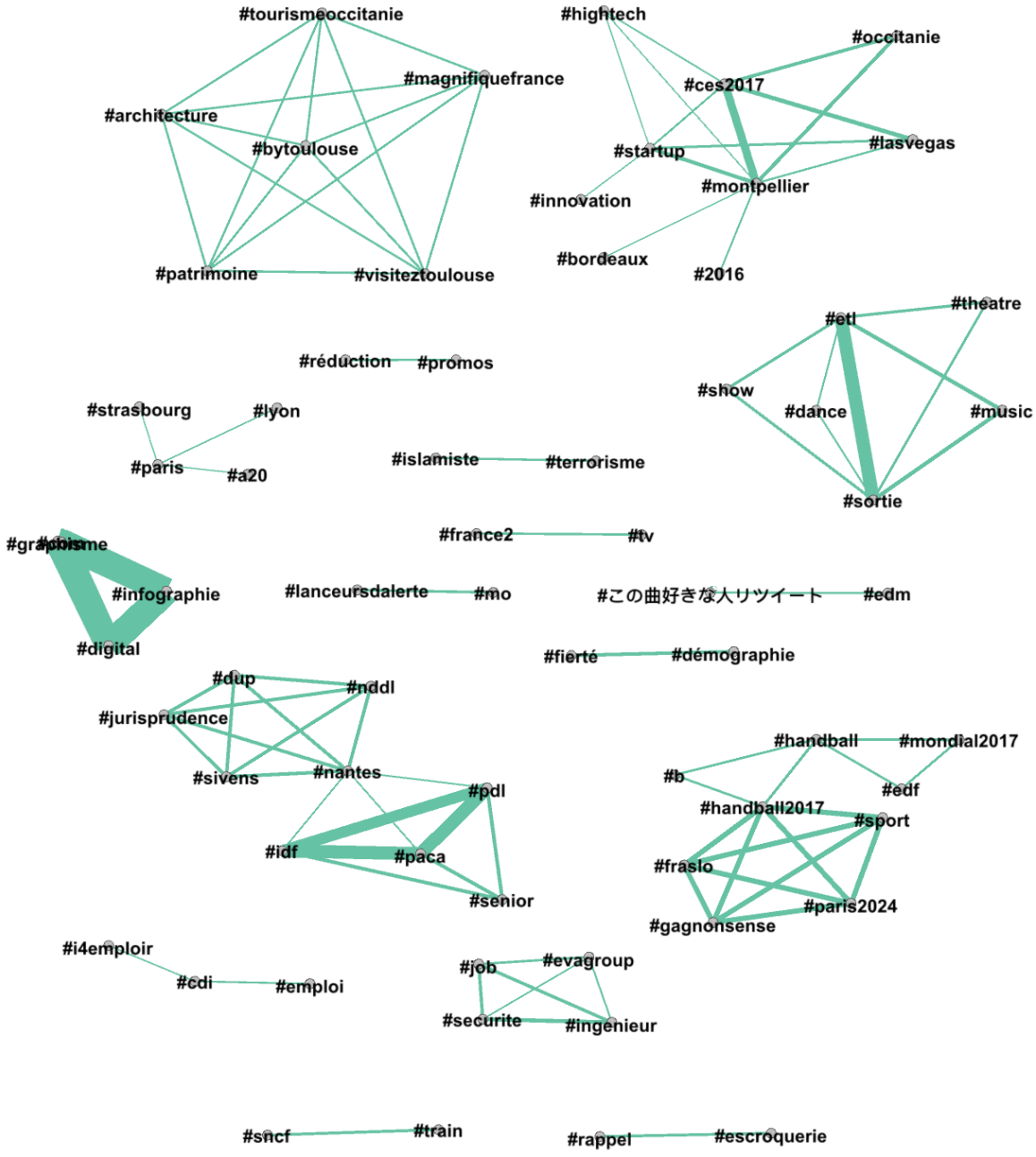

FIGURE 4 – Réseau de co-hashtags présents dans les tweets mentionnant #Toulouse

```
python3 dataset_treatments.py dataset_#Toulouse_5000.json -d days
Days,nb_tweets
1,447
2,1528
3,1534
4,1391

python3 dataset_treatments.py dataset_#Toulouse_5000.json -d hours
Hours,nb_tweets
0,41
1,26
2,23
3,32
4,25
5,66
6,135
7,267
8,255
9,343
10,326
11,328
12,238
[...]

python3 dataset_treatments.py dataset_#Toulouse_5000.json -d dates_days
Dates,nb_tweets
2017-01-04,1534
2017-01-03,1528
2017-01-02,447
2017-01-05,1391

python3 dataset_treatments.py dataset_#Toulouse_5000.json -d dates_minutes
Dates,nb_tweets
2017-01-03 14:22,1
2017-01-04 06:47,1
2017-01-04 11:21,1
2017-01-05 17:24,1
2017-01-03 15:31,3
2017-01-03 15:30,8
2017-01-04 12:14,1
2017-01-04 19:51,2
2017-01-03 10:59,1
[...]
```

FIGURE 5 – Exemples d'agrégation des tweets mentionnant #Toulouse par date