# Rapport de stage

Baptiste Fontaine

21 mai 2012

## Table des matières

# Introduction

Ce stage m'a été proposé par Christophe Prieur alors que je ne trouvais pas de matière pour valider mon UE libre [1]. Il s'est déroulé pendant la quasi-totalité du semestre, durée pendant laquelle j'ai principalement travaillé chez moi, avec un rendez-vous hebdomadaire avec C. Prieur pour avoir un suivi régulier. J'ai aussi pu travailler directement au LIAFA plusieurs fois pendant les vacances de Pâques.

C. Prieur et Stéphane Raux étudient la façon dont interagissent les utilisateurs de Twitter, et cherchent à expliquer les motivations des utilisateurs qui "retweetent" (*i.e.* qui re-postent du contenu publié par d'autres utilisateurs, et donc qui contribuent à sa dissémination sur le réseau). Pour ce faire, ils ont extrait des ensembles d'utilisateurs du réseau en fonction d'une URL que chacun de ces utilisateurs avait "tweeté". Chacun de ces ensembles est décrit par plusieurs graphes — $G_0$, $G_f$, $G_{RT}$ — dont les définitions sont données plus loin.

## Définitions

**Twitter :** Réseau social créé en 2006 sur lequel les utilisateurs peuvent publier de courts messages publics, et s'abonner aux messages d'autres utilisateurs. Ces messages sont appelés des "*Tweets*", et "*suivre*" quelqu'un signifie être abonné à ses *Tweets*. Lorsqu'un utilisateur poste un message à propos d'une information ou qui contient une URL, on dit qu'il *tweet* cette information ou URL. Le site propose une API pour les développeurs, permettant de récupérer toutes les informations sur les *Tweets* et les utilisateurs facilement. La version Web de Twitter est d'ailleurs basée sur sa propre API.

**Retweet :** *Tweet* d'un utilisateur qui est re-publié par un autre utilisateur, afin que ses abonnés voient le *Tweet* originel. Le verbe associé est "*retweeter*". Lorsqu'on *retweet* le *Retweet* d'un utilisateur, notre message est affiché comme étant le *Retweet* du *Tweet* originel. Ainsi, si l'utilisateur B retweet A, puis C retweet B, le *Tweet* de C est affiché comme étant un *Retweet* de A, sans mention de l'utilisateur B.

**$G_0$ :** Graphe dont les noeuds sont les utilisateurs qui ont tweeté une URL donnée, et les liens les interactions [2] qu'ils ont eu entre eux avant le début de la diffusion de l'URL. Dans les figures, un lien de $G_0$ sera représenté en rouge.

**$G_f$ :** Graphe des abonnés ("*followers*"), *i.e.* dont chaque lien matérialise une relation d'abonnement sur le réseau ; Un lien de A vers B indique que A suit B. Il est censé représenter les liens d'abonnements entre les utilisateurs

1. Unité d'Enseignement hors cursus au choix, à suivre pendant un semestre
2. Une interaction est matérialisée soit par un *Retweet*, soit par la mention du nom d'un autre utilisateur dans un *Tweet*

avant le début de la diffusion de l'URL [3]. Dans les figures, un lien de $G_f$ est représenté en bleu.

$G_{RT}$ **:** Graphe des *Retweets*, *i.e.* dont chaque lien matérialise un *Retweet* ; un lien de A vers B indique que A a reweeté B. Chacun de ces *Retweets* contient l'URL étudiée. Dans les figures, un lien de $G_{RT}$ est représenté en noir.

# Objectifs

Lorsque j'ai commencé le stage, S. Raux avait déjà extrait les informations pour 124 ensembles d'utilisateurs. Pour chacun de ces ensembles était disponibles un fichier de graphe $G_0$ et deux fichiers au format JSON, l'un étant simplement le résultat de la recherche effectuée sur une URL via l'API, l'autre une liste datée des *Retweets* entre ces utilisateurs, permettant de construire $G_{RT}$.

Mes objectifs étaient les suivants :

- Dans une première partie, récupérer les informations concernant les abonnements des utilisateurs, afin de générer les $G_f$.
- Dans une seconde partie, extraire des informations chiffrées sur les graphes générés, afin de permettre une étude statistique de ceux-ci.

# Méthodologie

Durant tout le stage, j'ai utilisé des scripts en **Ruby** pour automatiser les tâches répétitives et/ou trop longues à faire à la main. J'ai aussi utilisé le logiciel **Gephi** pour visualiser les graphes, et **R** pour les statistiques.

## Technique

Afin de représenter des graphes simplement, j'ai utilisé des listes de *hashs* [4] pour les noeuds et les liens de chaque graphe. J'ai utilisé le format *GDF* pour stocker les graphes, car c'est un format qui a l'avantage d'être peu verbeux (deux lignes plus une ligne par noeud ou lien) et facile à générer et à parser. Pour l'occasion, j'ai mis en ligne une petite bibliothèque sous forme de gem [5], `graphs`, permettant de manipuler des graphes et de les stocker au format *GDF* [6].

3. Les données relatives aux abonnements ont en fait été collectée après.
4. équivalent en Ruby des dictionnaires de Python ou des *HashMaps* de Java
5. paquet logiciel utilisé pour partager des modules en Ruby, équivalent des *eggs* de Python
6. La *gem* a été téléchargée plus de 400 fois depuis sa mise en ligne

### Première partie

Le principal obstacle pour la première partie était la limite horaire du nombre de requêtes à l'API de Twitter[7]. En effet, l'ensemble des graphes représente 7000 utilisateurs uniques ; récupérer les identifiants de leurs abonnés demande une requête par tranche de 50000[8]. J'ai écris un script pour récupérer ces informations, et S. Raux m'a proposé de le faire tourner en continu sur un serveur chez Linkfluence, ce qui a permis de récupérer les informations sous 48h. Le script générait un fichier par utilisateur listant ses abonnés. J'ai ensuite croisé ces listes avec les informations des graphes pré-existants pour générer les $G_f$.

### Seconde partie

Pour la seconde partie, j'ai dû rajouter des fonctions à ma petite bibliothèque afin de pouvoir faire de l'arithmétique sur les graphes : unions, intersections, *XOR*. Ainsi, générer l'intersection de deux graphes devenait aussi simple que la ligne de code suivante :

```
GDF::load("graph1.gdf") & GDF::load("graph2.gdf")
```

J'ai dû aussi gérer certains cas où les attributs des noeuds entre différents graphes n'étaient pas les mêmes, il fallait dans ce cas n'effectuer des comparaisons que sur les attributs communs des noeuds.

En utilisant ces fonctions, j'ai pû générer des pourcentages de recouvrement entre certaines parties de graphes (par exemple la proportion de $G_{RT}$ qui est dans $G_0$, ou la proportion de $G_{RT}$ qui est dans $G_f$ mais pas dans $G_0$, etc).

Pour terminer, il a fallu calculer les taux d'explications possibles pour les *Retweets*. Par exemple, un *Retweet* peut avoir eu lieu parce que la personne qui a retweeté a déjà interagi avec la source du *Tweet*, dans ce cas, un lien de A vers B dans $G_{RT}$ existe aussi dans $G_0$. Cela peut aussi être parce que A a déjà interagi avec quelqu'un qui a retweeté B auparavant, ou parce que A suit B tout simplement, ou encore parce que A suit quelqu'un qui a retweeté B auparavant.

## Résultats

L'ensemble des chiffres donnés ici concernent un ensemble de 124 graphes[9], 7215 utilisateurs uniques, et 9947 *Tweets* (dont 4446 *Retweets*).

7. les requêtes sont limitées à 350 par heure pour une application authentifiée
8. seuls une vingtaine de comptes dépassaient ce seuil
9. Chaque graphe étant un ensemble d'utilisateurs qui ont tweeté une URL donnée. Les graphes existent en trois variantes : $G_0$, $G_f$ et $G_{RT}$.

## Recouvrements

Pour chaque graphe, quatre intersections ont été calculées, avec à chaque fois le taux de recouvrement par rapport à $G_{RT}$.

| Intersection | Taux de recouvrement par rapport à $G_{RT}$ | | | |
|---|---|---|---|---|
| | premier quartile | médiane | moyenne | troisième quartile |
| $G_0$ et $G_{RT}$ | 60.28% | 72.22% | 70.85% | 82.85% |
| $G_f$ et $G_{RT}$ | 11.50% | 41.50% | 42.29% | 72.06% |
| $G_0$ et $G_f$ et $G_{RT}$ | 7.55% | 34.74% | 34.28% | 56.73% |
| $G_f$ et $G_{RT}$ sans $G_0$ | 0% | 4.37% | 8.00% | 12.33% |

FIGURE 1 – Taux de recouvrement des graphs

On remarque ainsi que 70.85% [10] des *Retweets* sont faits par des gens qui avaient déjà interagi avec l'auteur du *Tweet*, mais que seulement 41.5% [11] sont issus de personnes abonnées à celui-ci, ce qui montre bien qu'il y a eu une dissémination du *Tweet* originel sur le réseau, en dehors de ses propres abonnés. Le faible taux de recouvrement entre $G_f$ et $G_{RT}$ sans $G_0$ (8% en moyenne) confirme que très peu de *Retweets* sont fait par des gens qui suivent mais n'avaient pas interagi avec l'auteur du *Tweet*.

## Différentes explications possibles

Quatre explications différentes pour un *Retweet* étaient envisagées :

– l'utilisateur qui retweet a déjà interagi avec l'auteur du *Tweet* originel, autrement dit il existe un lien direct dans $G_0$.

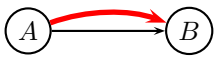


FIGURE 2 – Explication 1

– l'utilisateur qui retweet a déjà interagi avec quelqu'un qui a déjà retweeté le même *Tweet*, mais n'a jamais interagi avec l'auteur originel.
– l'utilisateur qui retweet suis l'auteur du *Tweet* originel, autrement dit il existe un lien direct dans $G_f$.
– l'utilisateur qui retweet suis quelqu'un qui a déjà retweeté le même *Tweet*, mais ne suit pas l'auteur originel.

Les différents scénarios explicatifs ont été étudiés indépendamment les uns des autres.

10. En moyenne
11. idem

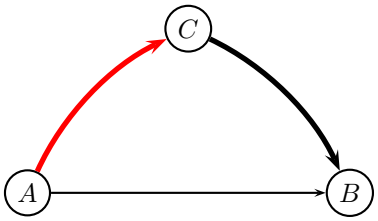


FIGURE 3 – Explication 2

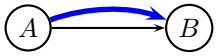


FIGURE 4 – Explication 3

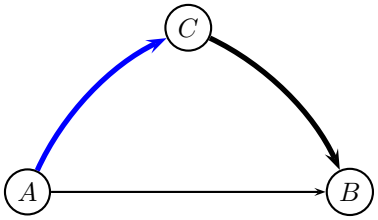


FIGURE 5 – Explication 4

### Interaction préexistante

Dans le premier scénario, un utilisateur retweet quelqu'un avec qui il a déjà interagi [12]. C'est celui le plus souvent rencontré ; en moyenne, 70.8% des *Retweets* sont issus de personnes qui avaient déjà interagi avec l'auteur du *Tweet*. La médiane est à 72.2%, le premier quartile à 60.2% et le troisième à 82.8%.

### Interaction indirecte

Dans le deuxième scénario, un utilisateur A a déjà interagi avec un autre, B, qui a retweeté l'auteur du tweet originel, C ; A n'a pas eu d'interaction directe avec C. On explique donc son *Retweet* par le fait qu'il a vu le *Retweet* de B avant. En moyenne, 24.9% des *Retweets* sont expliqués par ce scénario. La médiane est à 17.1%, les premier et troisième quartiles à 5.5% et 41.6% respectivement.

### Abonnement préexistant

Le troisième scénario est similaire au premier, sauf qu'ici, on n'explique pas le *Retweet* par une interaction mais par un abonnement : A retweet B parce qu'il

12. Ce qui correspond à un recouvrement entre $G_0$ et $G_{RT}$

est abonné à ses *Tweets*. En moyenne, 42.3% des *Retweets* sont expliqués par ce scénario. La médiane est à 41.5%, les premier et troisième quartiles à 11.5% et 72.0% respectivement.

### Abonnement indirect

Le quatrième scénario est similaire au premier, sauf que, comme dans le troisième, on n'explique pas ici le *Retweet* par une interaction mais par un abonnement [13] : A est abonné à B mais pas à C, B retweet C *puis* A retweet C *via* B. A a donc vu le *Tweet* originel de C retweeté par B auquel il est abonné, avant de lui-même le retweeter. Ce scénario explique 13.2% des *Retweets* en moyenne. La médiane est à 7.1%, les premier et troisième quartiles à 0% et 17.6% respectivement.

### Résumé

| Scénario | premier quartile | médiane | moyenne | troisième quartile |
|---|---|---|---|---|
| Interaction préexistante | 60.28% | 72.22% | 70.85% | 82.85% |
| Interaction indirecte | 5.56% | 17.16% | 24.96% | 41.66% |
| Abonnement préexistant | 11.50% | 41.50% | 42.29% | 72.06% |
| Abonnement indirect | 0.00% | 7.14% | 13.23% | 17.68% |

FIGURE 6 – Résumé des explications

## Conclusion

Ce stage m'a permis de mettre en pratique les connaissances apprises en cours ou par moi-même [14], de façon autonome, et avec la présentation de résultats. Il m'a permis de travailler avec des graphes, ce qui est (entre autres) l'objet du cours d'algorithmique en L3, que je suivrai l'année prochaine. J'ai aussi pu avoir un aperçu de ce que peut être la recherche en informatique.

13. Ce qui correspond à un recouvrement entre $G_f$ et $G_{RT}$
14. Le langage Ruby, par exemple