



Julien Topçu @JulienTopcu Sat Apr 01 01:42:18 +0000 2023

#Twitter vient de rendre open-source son algorithme, outre les blagues que je vais citer en thread, j'ai trouvé un truc intéressant dans le commit initial et qui a été supprimé par la suite. En suivra une vulgarisation de l'algo #thread ■■■1/
<https://t.co/F55L7TRr5q> <https://t.co/a4CUKSW9Ca>

```
val featureMap = FeatureMapBuilder()
    .add(AccountAgeFeature, query.getOptionalUserId.flatMap(SnowflakeId.timeFromId0pt))
    .add(ClientIdFeature, query.clientContext.appId)
    /**
     * These author ID lists are used purely for metrics collection. We track how often we are
     * serving Tweets from these authors and how often their tweets are being impressed by users.
     * This helps us validate in our A/B experimentation platform that we do not ship changes
     * that negatively impacts one group over others.
     */
    .add(DDGStatsDemocratsFeature, ddgStatsAuthors.longSeq(Democrats).toSet)
    .add(DDGStatsRepublicansFeature, ddgStatsAuthors.longSeq(Republicans).toSet)
    .add(DDGStatsVitsFeature, ddgStatsAuthors.longSeq(Vits).toSet)
    .add(DDGStatsElonFeature, ddgStatsAuthors.longValue(Elon))
    .add(DeviceLanguageFeature, query.getLanguageCode.map(getLanguageISOFormatByCode))
    .add(
        GetInitialFeature,
        query.pipelineCursor.forAll(cursor => cursor.id.isEmpty && cursor.gapBoundaryId.isEmpty))
    .add(
        GetMiddleFeature,
        query.pipelineCursor.exists(cursor =>
            cursor.id.isDefined && cursor.gapBoundaryId.isDefined &&
            cursor.cursorType.contains(CasCursor)))

(
    "author_is_elon",
    candidate =>
        candidate
            .getOrElse(AuthorIdFeature, None).contains(candidate.getOrElse(DDGStatsElonFeature, 0L)),
    (
        "author_is_power_user",
        candidate =>
            candidate
                .getOrElse(AuthorIdFeature, None)
                .exists(candidate.getOrElse(DDGStatsVitsFeature, Set.empty[Long]).contains)),
    (
        "author_is_democrat",
        candidate =>
            candidate
                .getOrElse(AuthorIdFeature, None)
                .exists(candidate.getOrElse(DDGStatsDemocratsFeature, Set.empty[Long]).contains)),
    (
        "author_is_republican",
        candidate =>
            candidate
                .getOrElse(AuthorIdFeature, None)
                .exists(candidate.getOrElse(DDGStatsRepublicansFeature, Set.empty[Long]).contains)),
```

En gros ce que je comprends de ces bouts de code, les équipes A/B testaient des features et évaluaient si cela impactait négativement les stats en terme de reach d'Elon Musk. A priori ça checkait aussi les différences de scores entre les républicains et les démocrates ■■■2/

Le code est en GNU Affero General Public License v3.0 ce qui veut dire que techniquement on peut réutiliser leur algo pour un usage commerciale si et seulement si on rend entièrement publique le code source qui l'utilise ■■■3/

Le Readme est assez succinct, il nous invite à contribuer mais il n'y a aucune commande documentée pour builder les sources ■ (5 langages a priori et Bazel en tant que builder). ■■■4/

Et la blague c'est qu'il n'y a aucun tests (à part des tests de charge à priori), ce qui veut dire que pour contribuer c'est techniquement... très compliqué ■■■5/

Les équipes ont cependant décrit l'algo de recommandation ici dans un blog : https://blog.twitter.com/engineering/en_us/topics/open-source/2023/twitter-recommendation-algorithm ■■■6/

A priori les tweets de votre TL viendraient à 50% de gens que vous suivez et 50% de gens que vous ne suivez pas.

Chez les personnes que vous suivez, ils utilisent la technique du Real Graph.

<https://www.ueo-workshop.com/wp-content/uploads/2014/04/sig-alternate.pdf>

■7/

Cette technique évalue votre probabilité interagir avec quelqu'un.

A priori parmi ces paramètres il y a le fait que le numéro de téléphone de la personne que vous suivez soit dans votre carnet d'adresse de votre téléphone (ou bien votre carnet d'adresse mail) ! ■8/ <https://t.co/hKJfVEKUor>

2.1 Graph Generation

There are three ways for an edge from user *A* to user *B* to be added to the RealGraph in any given time period: (i) if *A* follows *B*, (ii) if *B* is in *A*'s phone or email address book (again, only if the requisite permissions have been granted by the users), and (iii) if *A* interacted with *B*. Currently, we compute the RealGraph on a daily basis, so we add these edges daily. Then, we merge them with the previous day's

¹We only add address book edges where *both* ends of the edge have allowed Twitter permission to use this information.

Bien-sûr sauf si vous avez révoqué cette permission sur Android ou iOS...

Puis ça va regarder vos interactions explicite avec l'utilisateur (likes, RT...) ou implicite (vue du profil, lecture d'un tweet, clique sur un lien). 9/

Ca explique clairement pourquoi je me retrouve avec des boosts de reco sur certaines personnes après avoir cliqué sur un tweet...

Puis ça va pondérer les interaction (un espèce de score) en prenant en compte par exemple le nombre de jour qui s'est écoulé depuis l'interaction 10/

Le nombre de jours depuis une interaction du même type (par exemple un autre like), le nombre de jours depuis la première fois ce type d'interaction etc... 11/

Ca va ensuite utiliser une régression logistique, et plus le score de cet utilisateur est grand vis à vis de vos interactions, plus vous verrez de ses tweets dans votre TL (ne cliquez pas n'importe où :p) 11/

Chez les gens que vous ne suivez pas (soit 50% de votre TL), ça va regarder quels sont leurs tweets avec lesquels les gens que vous suivez interagissent.

Ca estime aussi qui a tendance à aimer des tweets similaires à ceux que vous likez. 12/

A partir de là ils vont sélectionner des tweets potentiels à vous recommander.

Ils utilisent donc 2 approches. La 1ère méthode est nommée GraphJet pour donner un score à ces tweets avec lesquels vous interagissez 13/

<https://www.vldb.org/pvldb/vol9/p1281-sharma.pdf>

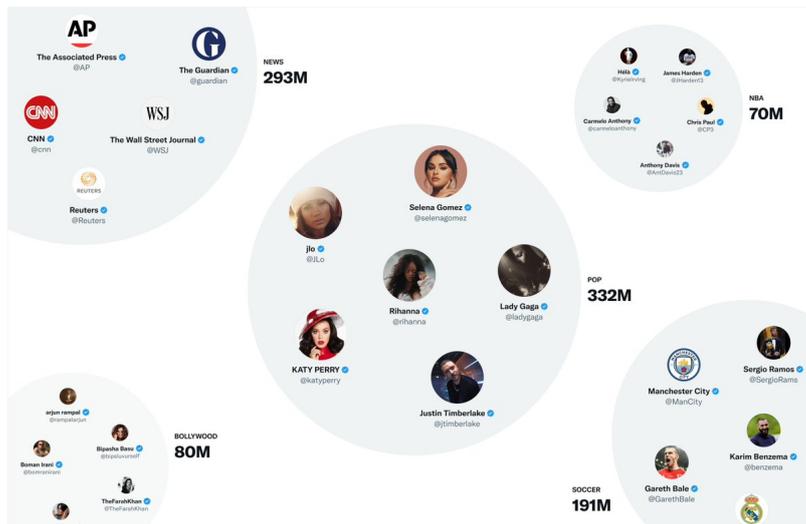
GraphJet maintiens une cartographie temps réel des interactions des utilisateurs les tweets de la plateforme sur les n dernières heures

Chaque serveur GraphJet ingère jusqu'à un million d'intérations par seconde et calcule jusqu'à 500 recommandations par seconde 14/

La deuxième méthode est une méthode de clustering, où ils regroupent les tweets et les utilisateurs qui ont des intérêts similaires aux vôtres.

On est donc tous regroupés par thèmes comme suit, et les tweets les plus populaires de nos clusters nous sont

recommandés 15/ <https://t.co/MjiiqTTgmH>



Bon ensuite tout ça, ça fait beaucoup de tweets. Ils vont donc calculer un ranking pour les ordonner dans votre timeline, un peu comme ce que fait google quand vous faites une recherche pour déterminer dans quel ordre vous présenter les résultats. 16/

Avec les tweets venants de vos follows et en dehors de vos follows, ils ont environ 1500 tweets candidats à être recommandés dans votre TL.

A priori avant ranking, "tous les tweets sont traités de manières équitables peut importe de là où ils viennent"

Sauf si c'est Elon■? 17/

Pour le ranking, ils vont filtrer les comptes que vous avez bloqués ou mute (marrant que ce ne soit fait pas plus en amont d'ailleurs), la diversité des auteurs pour éviter d'être spammé par la même personne (Sauf si c'est Elon■?) puis 18/ un équilibre entre le nombre de tweets provenant de vos follows et hors follows est assuré.

Cela va évaluer aussi votre "fatigue" vis à vis d'un tweet. Il parle de feedback négatif de votre par sans précision

19/

Ils vont exclure les tweets hors de vos follows, si aucun de vos follows directs n'a interagit avec, ou si aucun de vos follows ne suit l'auteur du tweet (ne suivez pas n'importe qui :p) 20/

Ils vont inclure aussi des tweets de contexte, notamment le tweet parent si c'est une conversation.

Ils vont vérifier si le tweet à recommander n'a pas été édité, et si c'est le cas, ils vont vous fournir la version la plus récente 21/

Et pour finir, ils vont procéder au "Home Mixer", où ils vont mélanger ces recommandations avec des pubs, des recommandations de profils à suivre...

Et c'est tout pour la vulgarisation ! 22/22

Ce qu'ils appellent les feedbacks négatifs (sur le fait d'être recommandé dans les TLs) ce sont : le nombre de gens qui vous ont bloqué, le nombre de gens qui vous ont mute, si vous avez fait des signalements abusifs et si vous spammez les signalement.

https://github.com/twitter/the-algorithm/blob/main/src/scala/com/twitter/interaction_graph/scio/agg_negative/Interaction

Je viens de voir aussi ce thread complémentaire

A priori votre ratio following/followers jouent aussi, ainsi que le fait d'être Twitter Blue (ça en s'en doutait)

https://twitter.com/steventey/status/164189238656460768?t=pojNpjjxTq8w6_PuZaFAfg&s=09