RENEWAL

A real-time evaluation platform for news recommender systems

Julien Hay, Bich-Liên Doan, Fabrice Popineau

News recommendation is a specific task in the area of recommender systems because of both the nature of items (news volatility, dynamic popularity, textual content, etc.) and the need to evaluate recommendation algorithms in real-time. **Challenges** are a fun way to stimulate research. We propose a platform called **Renewal** to host a news recommandation challenge. The platform provides the **evaluation service** for contenders programs submitted by research teams. To our knowledge, this platform is the only one which offers a user application fully dedicated to the **cross-website** and **cross-language** news articles recommendation task. It also offers a large panel of **context / demographic clues** and a **long-term user history** through a dedicated mobile app.

Users' mobile app

What data do we collect?

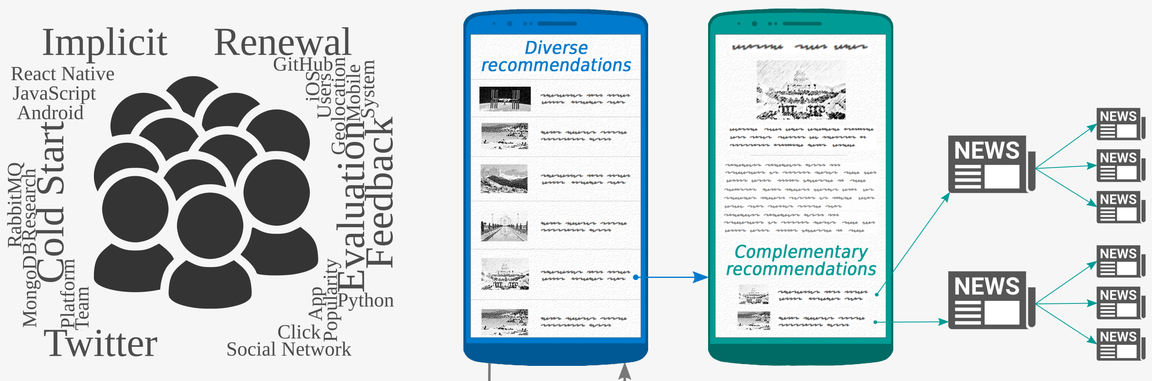
- **Contextual data**: geolocation, battery level, network speed...
- **Behavior analysis**: click, scroll, read time, share, item deletion...
- **Twitter** and **Facebook** connection providing social network data

Why?

- To provide **rich clues** to recommender systems
- To **analyse** recommender systems performances
- Positive and negative **implicit feedback** through rich behavior analysis (read time and scrolling)
- Against **CTR** (Click-Through Rate) metric which considers a click as the main indicator

The Renewal mobile app

- The **central concept** is built around 2 kinds of recommendations : diverse recommendations on the main page and complementary recommendations below each news article as the main *NewsREEL* task [1]
- We use the **React Native** framework for a responsive UI



Diverse recs

- The app recommends **diverse** and **serendipe** news items [2]
- Users scroll down and choose a news article to read
- **Auto preloading** of recommendation lists

Complementary recs

- Recommends news items **complementary** to the current news article
- **Non-reduntant** and providing **additional** informations
- Browse over news articles as a **tree structure**, from one complementary news to others

The Renewal platform

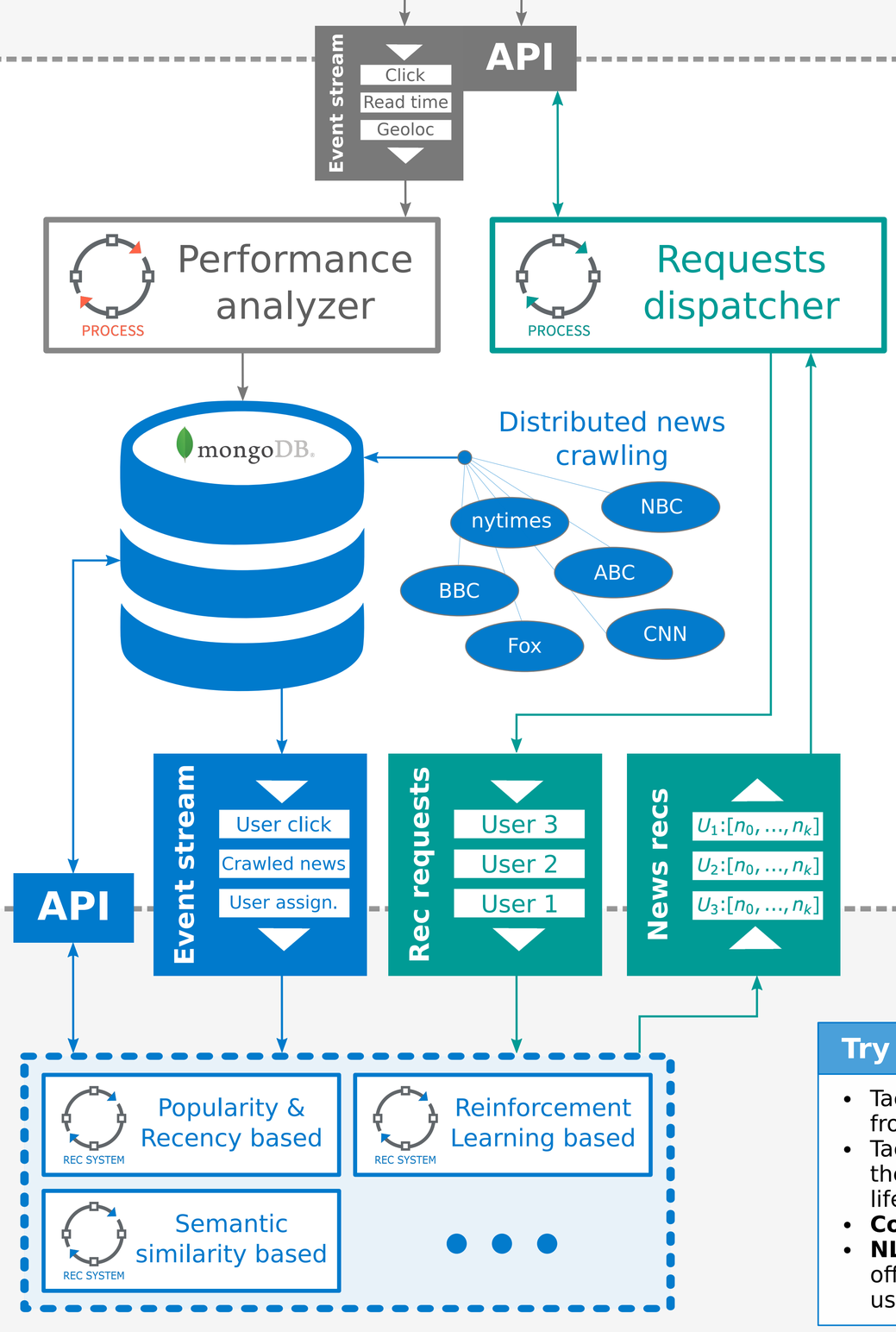
Online evaluation

- We provide the opportunity to evaluate algorithms on **online settings** which measure the true **user satisfaction** [2]
- We analyse user behaviors to provide a **real-time performance analysis**
- All **statistics** are displayed on www.renewal-research.com

Renewal platform's properties

- A **distributed** architecture for indexing
- A subset of all users is assigned to each recommender system to relieve the **user profile modeling** (scalability issues).
- Given n as the number of users, a as the number of recommender systems per user and m the total number of recommender systems, y the number of **assigned users per system** is :

$$y = \frac{na}{m}$$



Evaluation stategy

- The evaluation is based on **A/B testing**
- Users are **assigned** to each recommender system on a **weekly basis**
- When the mobile app requests lists of recommendations, the request dispatcher module **redirects the request** to all affected systems
- The performance analyser gets events from the event stream and computes the **positive feedback rate**

What data do we index?

- **News articles** from Multiple RSS feeds and other sources
- **User data** from social networks and the Renewal mobile app
- **User behaviors**

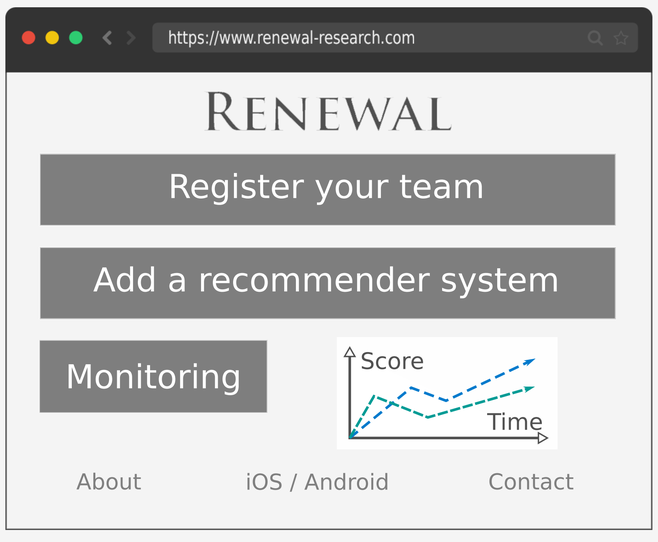
Teams' recommender systems

Get started

1. Subscribe on **www.renewal-research.com**
2. Download our **baselines** in Java / Python / C++ available on GitHub to learn basic usage
3. Design **your own recommender system** and deploy it on your own machines
4. Watch **performance curves** in real-time and compare your algorithms to others
5. **Contribute** to the research in the recommender system field by writting papers showing your results

Systems' properties

- **Unavailability tolerance:** your system is not server-oriented, instead, you just need to access the API and your dedicated queues using credentials
- Recommendation are **preloaded** in mobile apps, so you don't have a short response time constraint, it prevents **scalability issues**



Try your own algorithms

- Tackle the **user cold start** using user data from social networks [3]
- Tackle the **item cold start** which is specific to the news rec task due to the recency / short lifespan of news constraints [4]
- **Content-based filtering** becomes relevant
- **NLP** methods or a diversity / popularity trade-off using **reinforcement learning** could be used

How to recommend news articles?

1. Use the **event stream** to grab news articles and user behavior
2. Use the **API** to request news content, popularity information, user data...
3. Receive a recommendation request from a user by taking it from the **rec requests queue**
4. Send your recommendation lists (which is a subset of news articles IDs) in the **news recs queue**
5. You can **precompute** your recommendations, or just **compute** it directly when you receive a recommendation request, you can also **update** your recommendation lists periodically

Our plans and perspectives

We plan to launch the platform in **late 2018**. A 1 month evaluation will be organized as part of an **evaluation campaign** during the year 2019. We also plan to organize specific challenges, e.g. on the recommendation **explanation**, or by filtering a **subset of users** (those who filled in their demographic infos, those linked to social networks...). Given the mobile app and registered users, there are many oportunities for research purposes: **crowd-labeling** (fake news detection, quality, readability, writing style...), integrating a **personality** survey to try personality-based recommendation algorithms...

References

[1] Kille, B., Lommatzsch, A., Hopfgartner, F., Larson, M., & Brodt, T. (2017). CLEF 2017 NewsREEL overview: Offline and online evaluation of stream-based news recommender systems.
[2] Beel, Joeran & Langer, Stefan. (2015). A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems.
[3] Son, Le. (2014). Dealing with the new user cold-start problem in recommender systems: A comparative review. Information Systems.
[4] Gulla, J.A., Marco, C., Fidjestøl, A.D., Ingvaldsen, J.E., & Özgöbek, Ö. (2016). The Intricacies of Time in News Recommendation.

Acknowledgements

We want to thank Alexis Jodar and Fabien Michel who have made a significant contribution to the development of the platform. This work is funded by Octopeek SAS and the LRI.