# PEN recsys: a Personalized News Recommender Systems Framework

Florent Garcin　　Boi Faltings

Artificial Intelligence Lab
Ecole Polytechnique Fédérale de Lausanne
Switzerland
firstname.lastname@epfl.ch

## ABSTRACT

We present the Personalized News (PEN) recommender systems framework, currently in use by a newspaper website to evaluate various algorithms for news recommendations. We briefly describe its system architecture and related components. We show how a researcher can easily evaluate different algorithms thanks to a web-based interface. Finally, we discuss important factors to take into account when conducting online evaluation, and report on our experience when deploying recommendations on a live-traffic website.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

## Keywords

recommender system, news, online evaluation

## 1. INTRODUCTION

Researchers in the recommender systems community have developed open-source libraries which try to bring a growing number of recommender algorithms under one roof [6, 5, 1, 13, 15, 19]. Most of these libraries are designed for research purposes to conduct offline evaluations and only a few target online evaluations on production websites [19, 15].

In this paper, we are interested in *online* evaluation of state-of-the-art algorithms for *news* recommendations. Unfortunately, it is not possible to use current open-source platforms because they are not tailored to the specific needs of news recommendations and thus are difficult to adapt to the news domain [7].

The most important challenge of news recommendation is that news items are evolving very quickly. Stories and topics emerge and vanish rapidly and outdated stories are no longer interesting. Hence the recommender system must take into account these changes.

To this end, we present the PEN recsys framework for online evaluation of news recommender systems. PEN recsys is designed with 4 criteria in mind. First, it has to be *fast*. The framework must provide real-time recommendations as soon as possible, without making the users wait. Second, it must be *reliable*. It is not acceptable for a newspaper website to suffer from crashes. Third, a *flexible* design is important. It should be easy to add new components or extend recommender systems. Finally, it must be *scalable*. News websites are subject to unpredictable visit peaks and the framework must be able to handle them by delivering recommendations on time and without problems.

Implementing a recommender system for production websites is challenging [14]. In the following, we describe our framework and explain our design choices. We discuss important factors to take into account when conducting online evaluation and report on our experience when deploying recommendations with a live-traffic website. The PEN recsys framework is currently in use by the news website swissinfo.ch[1] for evaluating various recommender systems.

## 2. RELATED WORK

Over the last few years, several libraries for recommendations have been developed. In the following, we present the ones that are still actively under development and free/open source.

*MyMediaLite* [6] is an actively maintained C# library. It contains simple baseline techniques such as slope-one or random/most popular item, several variants of $k$-nearest neighbour models and some matrix factorization methods. Unfortunately, MyMediaLite is not thread-safe, and thus is not suitable for concurrent access required in a highly-dynamic environment such as the news domain.

*Apache Mahout* [1] is a distributed machine learning library in Java. Although it is not specifically developed for recommendations, it contains some collaborative filtering algorithms. This library implements the Map/Reduce paradigm and is tailored for distributed systems. As we discuss in Section 6.2, standard news websites do not need such complex distributed framework because it brings an heavy overhead when deployed on a single server.

[1]www.swissinfo.ch

*LensKit* [5] is a Java-based library developed for offline research environments. It focuses on collaborative filtering techniques.

*GraphLab* [13] is a C++ collection of machine learning toolkits on graphs. One toolkit is dedicated to collaborative filtering with implementations of Alternating Least Squares, Stochastic Gradient Descent and Singular Value Decomposition.

*easyrec* [15] is a web service in Java generating recommendations through an API. *easyrec* is designed for online evaluation. However, it is not clear which algorithms are available. Recently, *easyrec* reached the final stage of its research project and it is also not clear whether this engine will stay free and open source or not.

*reclab* [19] provides a Java API to build recommender models. It does not come with any implemented algorithms, and it is designed to run together with a cloud service provided by the developers. Unfortunately, this library seems no longer active.

These libraries are all useful for research purposes but cannot be used on production websites due to restrictions on licenses for commercial usages. In addition, most of them are designed for offline evaluation.

Specific to news recommendations, the literature describing actual implementation of algorithms on production websites is scarce. Das et al. [4] give a short description of the system components for generating recommendations on the live system Google News. Li et al. [11] describe the SCENE framework aiming at news recommendations, but they evaluate their algorithm offline. Tavakolifard et al. [18] explain the implementation of a recommender engine for a mobile news application. Finally, Kirchenbaum et al. [10] conducted an online evaluation of several recommender systems for news articles.

In this work, we focus on describing a general framework to evaluate recommender algorithms for news in an online environment. We report on our experience in deploying such framework.

## 3. SYSTEM ARCHITECTURE

The PEN recsys framework is designed around 6 main components. Figure 1 gives a brief overview of these components and their interactions.

The *dispatcher* randomly assigns a recommender system to a user, performing A/B or multivariate testing. The *recsys* $1, 2, 3, ..., k$ are the different algorithms to evaluate (See Section 4 for more details). Some algorithms rely on click statistics. The component *statistics* gathers click statistics about the stories. Other recommender systems need the content of news articles, or more specifically its topic distribution.

The component *topic model* is in charge of keeping the topic model up-to-date. We use the Latent Dirichlet Allocation (LDA) method as probabilistic topic model [3, 9]. Building a topic model takes time and resources. We implemented the offline LDA [3] in which we build the topic model at regular intervals and based on the number of new items. This option is useful in the case of a small corpus of news items or when the number of new items per day is small. In the case of a larger corpus or when the number of new items per day is more important, we also implemented an online version [9] which is useful for news items arriving in a stream.

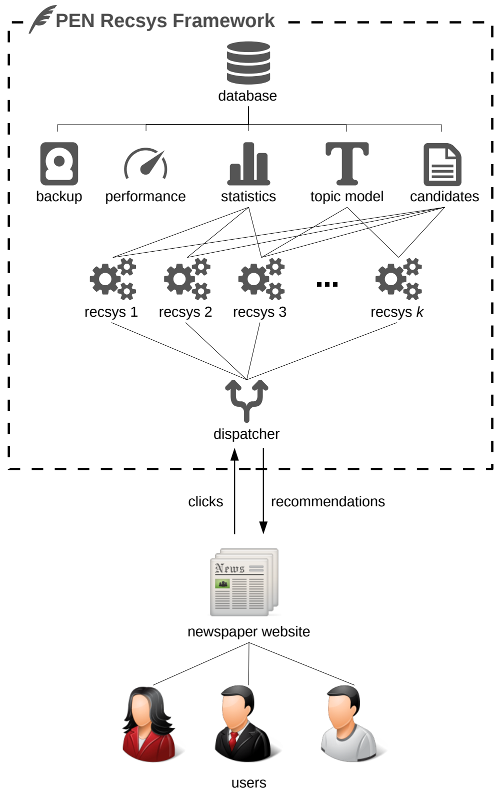


**Figure 1: System architecture and components**

Candidate items for recommendations are provided by the *candidates* component. This component periodically queries the news website for fresh, unseen items and keeps track of the clicked ones. Let $\mathcal{F}$ be the set of fresh items and $\mathcal{P}$ the set of clicked items such that $\mathcal{F} \cap \mathcal{P} = \emptyset$. We define the candidate set $\mathcal{C}$ as $\mathcal{C} = \mathcal{F} \cup \mathcal{P}$. It thus contains fresh items but also old ones. It is possible to control the size of $\mathcal{C}$ thanks to two parameters: the size of the fresh set $\mathcal{F}$ and the size of the clicked set $\mathcal{P}$. The clicked set contains the latest $|\mathcal{P}|$ clicked news items.

The *performance* component generates periodically performance reports of the algorithms under evaluation. The *database* stores the clicks, statistics and performance reports for offline analysis. It provides a simple abstraction so that algorithms can access useful information such as the user history, preference or item statistics.

Finally, the *backup* component periodically triggers backup to the hard disk of the various states of the system such as the current topic model and click statistics. This is useful if we want to roll back to a previous set of parameters.

It is important to deliver recommendations to the user as soon as possible. With that in mind, the platform is designed to reduce this latency to the minimum. Hence tasks that are not essential for generating recommendations are run in the background. For instance, the database is known to be a bottleneck. Thus statistics are first cached in memory and later stored in the database when resources are available.

When a reader clicks on a news item, the website sends to the PEN recsys a token id representing the reader and the respective item id. Note that the token id can be the reader id when the reader logs into the website or a completely anonymized string representing the current session of a non-logged reader. In our case, the website sends the latter.

The PEN recsys framework follows the software-as-a-service paradigm and is implemented using Java EE technologies. It scales very well since each component can be physically located on different sites.

## 4. RECOMMENDER SYSTEMS

The framework contains various algorithms such as the 4 versions of context-tree recommender systems [7], a simple collaborative filtering [17], a content-based approach [12], most popular articles, and random articles. We plan to implement more algorithms in the future.

To add a new algorithm, we just need to implement a single method `getRecommendations`. If required, the algorithm can have access to the click statistics, fresh news stories, or topic model via the specific components described in Section 3.

## 5. INTERFACES

The PEN recsys framework has a web-based control panel (Fig. 2) allowing the researcher to configure the general behaviour of the framework, enable/disable an algorithm or fine tune its parameters. In Figure 3, some recommender systems are enabled for an online multivariate evaluation (random articles, most popular recommender and VMM recommender [7]), and the specific options of the VMM recommender are displayed.

The researcher can also check the performance of the enabled algorithms. Figure 4 shows the performance panel with 3 metrics: success@k, mean average precision and the average clicks per visit.

## 6. DISCUSSION

Since July 2013, we are evaluating various recommender systems with live traffic on the news website *swissinfo.ch*. swissinfo.ch is a 10-language news website owned by the Swiss broadcasting corporation. Its content is produced specifically for an international audience with interests in Switzerland. swissinfo.ch gives priority to in-depth information on politics, society, business, culture and science & technology. It has about 1.7 million clicks per month.

Unfortunately at the time of writing, our evaluation is still being conducted and it is too soon to account on our results. However, we discuss here important factors that influence directly the recommendations, and report on our experience in developing and deploying the PEN recsys framework.

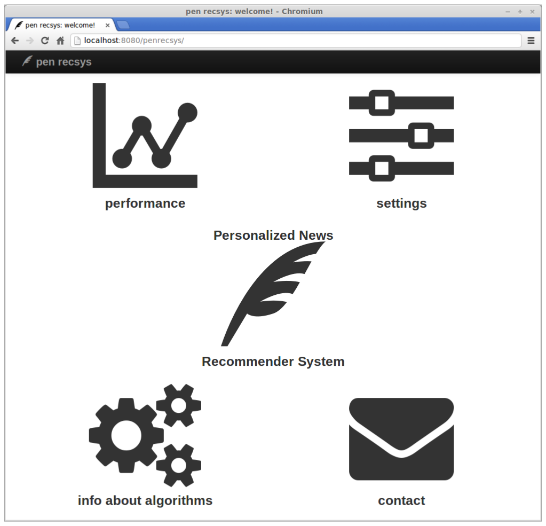
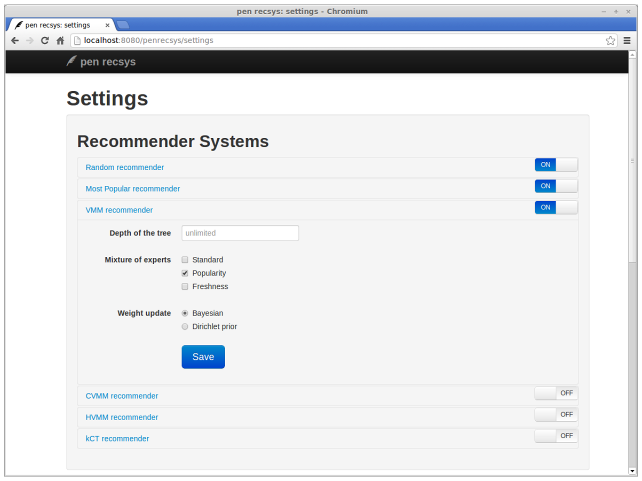


**Figure 2: screenshot of the main panel**


pen recsys: settings - Chromium
pen recsys: settings
localhost:8080/penrecsys/settings
pen recsys

Settings

Recommender Systems

Random recommender ON

Most Popular recommender ON

VMM recommender ON

Depth of the tree: unlimited

Mixture of experts: Standard; [✓ Popularity]; Freshness

Weight update: [✓ Bayesian]; Dirichlet prior

Save

CVMM recommender OFF

HVMM recommender OFF

kCT recommender OFF

**Figure 3: screenshot of the setting panel (partial)**

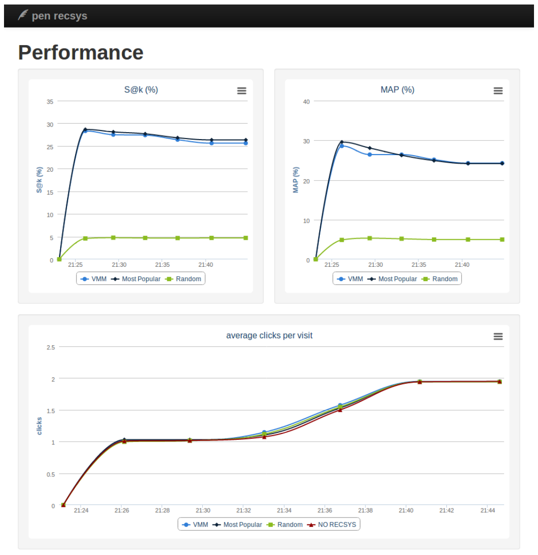
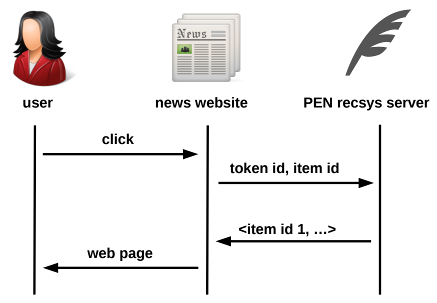
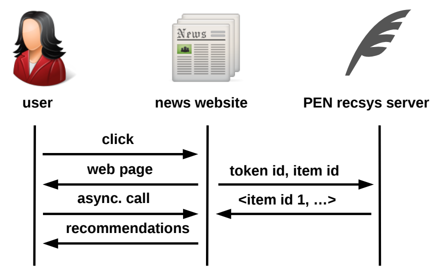


**Figure 4: screenshot of the performance panel**



(a) Synchronous



(b) Asynchronous

**Figure 5: Two possible solutions to deliver news recommendations.**

## 6.1 Evaluation

For offline settings, there are many ways to evaluate a recommender system [16]. In our previous works [8, 7], we measured the performance with respect to accuracy and novelty. For accuracy, we considered the Success@k and the mean average precision. Novelty is defined by the ratio of unseen and recommended items over the recommended items. In the context of news recommendations, novelty is essential because it exposes the readers to relevant items that she would not have seen by herself.

An offline evaluation is very artificial because it is not possible to assess the impact of recommendations on real users. In an online environment, the evaluation are motivated by different factors, and the way to measure the performance of a recommender system is different.

Most of the time, a news website is interested in generating revenues. These revenues come from either advertisements (ads) displayed on the website or paid articles (or sometimes both). In general for online advertisement, there are two revenue methods: pay per impression or pay per clicks. With the former, the news website receives monetary compensation for displaying ads while with the latter every time a user clicks on the ad.

In all cases, news website are incentivized in increasing page views. So one way to evaluate the performance of a recommender system is to look at the generated revenue from readers using this system.

swissinfo.ch is owned by a public non-profit organisation and financed by the Swiss government. It creates high-quality contents with no advertisements, and they do not have any subscription schemes or paid articles. Their motivation for implementing a recommender system is to bring an added-value to the readers but also to increase the number of clicks per visit.

For online settings, we measure the click rate on recommended items. More precisely, we compare simultaneously the average clicks per visit over the different algorithms.

In the PEN recsys framework, we decided to have both the offline and online performances such that we can study how a recommender algorithm behaves in both settings.

Previously, we conducted an offline evaluation of the algorithms [8, 7]. This allowed us to understand how the parameters influence the performance of the various algorithms, and pre-select the best parameters for the online evaluation. Since July 2013, we are evaluating various recommender systems and at the time of writing the evaluation is still on-going. Although it is too early to draw any conclusion, preliminary results show an increase of at least 20% in average clicks per visit when recommendations are displayed against no recommendations.

## 6.2 Real-time Recommendation and Latency

In a highly-dynamic domain such as news, providing *real-time* recommendations is crucial and challenging. Most of the standard recommender algorithms cannot be applied directly to the news domain [2, 8]. They are designed for product recommendations where updating a model once a day or week is acceptable. However, in our case the models need to be reactive and updated on-the-fly as fresh news stories come in.

We paid careful attention to optimize the PEN recsys framework in such a way that it generates recommendations very quickly. From a server-side perspective, it is important to remove any system bottleneck. For instance, storing information into a database is a known bottleneck. It is thus better to use caching methods which allow to keep information in memory first, and save them in a database when resources are available later. Moreover, the PEN recsys framework relies on threads and concurrent data structures. This allows to generate recommendations simultaneously to multiple readers.

A naïve solution to deliver a news content to the user is to use a synchronous approach. In Figure 5(a), the user clicks on a news item. Then, the swissinfo.ch's server queries the PEN recsys server. The PEN recsys generates recommendations and sends them back to the swissinfo's server, which encapsulates them in the web page. The swissinfo.ch's server sends the page with the recommendations back to the user. As a result, the user has to wait to see the content of the page until the recommendations are ready. This approach is not efficient.

We choose an asynchronous solution depicted in Figure 5(b). When a user clicks on a news item, swissinfo.ch's server sends the content of the page to the user's browser and at the same time queries the PEN recsys server. At that moment, the PEN recsys server generates recommendations and sends them back to the swissinfo.ch's server. The browser makes an asynchronous call, i.e. with Ajax technology, to the swissinfo.ch's server in order to fetch these recommendations. swissinfo.ch forwards the recommendations to the browser as soon as they are ready. If they are ready before the asynchronous call, the swissinfo.ch's server stores them temporarily. The advantage of this approach is that the original content of the page is not blocked while the recommendations are generated.

swissinfo.ch had another requirement: deliver recommendations within 1 second. When this threshold expires, no recommendations are displayed. So far, we never witnessed such behaviour.

For our online evaluation, the PEN recsys framework is running on a standard workstation with a dual-core CPU @ 2.6Ghz and 4GB of RAM. At the time of writing, the current memory usage is 48% with a CPU load of 51%. It handles visit peaks smoothly and recommendations are always delivered on time in less than 30ms (including connection overhead).

Consequently, we believe that standard news websites do not need an heavy system such as [1] but special cares must be taken by the researcher when designing and implementing the system.

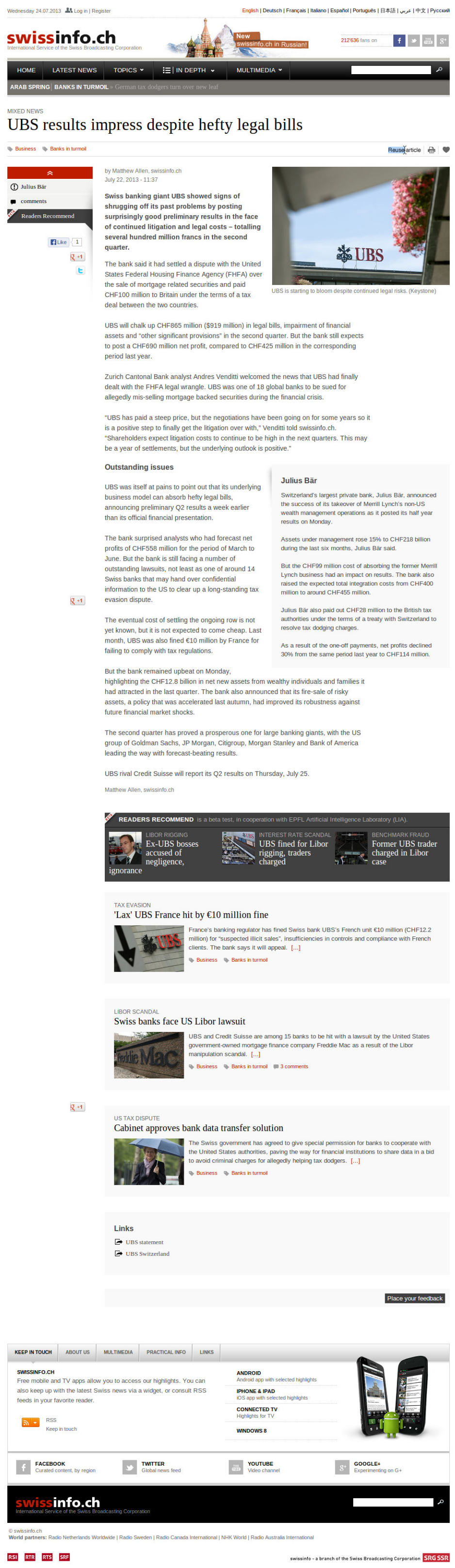
## 6.3 Presentation

A recommender system is a small piece of a bigger and complex environment. In our case, we have little to no control over the other elements of the environment such as the presentation layer.

swissinfo.ch's website has two types of news items: *stories* and *tickers*. The former are in-depth news articles written by swissinfo.ch's journalists. The editor-in-chief selects a topic of the day and the journalists produce relevant articles about the selected topic. The articles are translated in the 10 languages and modified to fit cultural backgrounds and differences. Figure 6 shows a screenshot of one story.

The second kind of items are *tickers*. Tickers are unedited short news items coming from press agencies. The number of tickers per day is very large, while the number of stories is significantly smaller.

The web page is split into several areas, sketched in Figure 8. The *header* has the menu buttons to browse the website and jump to different sections and topics, while the *footer* contains information about swissinfo.ch, links to social media sites, links to swissinfo.ch mobile applications and legal notices.


Wednesday 24.07.2013
Log in | Register
English | Deutsch | Français | Italiano | Español | Português | 日本語 | عربي | 中文 | Русский
swissinfo.ch
International Service of the Swiss Broadcasting Corporation
New swissinfo.ch in Russian!
212'636 fans on
HOME
LATEST NEWS
TOPICS
IN DEPTH
MULTIMEDIA
ARAB SPRING
BANKS IN TURMOIL » German tax dodgers turn over new leaf
Business
Banks in turmoil
Reuse article
Julius Bär
comments
Readers Recommend
Like 1
+1

MIXED NEWS

UBS results impress despite hefty legal bills

by Matthew Allen, swissinfo.ch
July 22, 2013 - 11:37

**Swiss banking giant UBS showed signs of shrugging off its past problems by posting surprisingly good preliminary results in the face of continued litigation and legal costs – totalling several hundred million francs in the second quarter.**

The bank said it had settled a dispute with the United States Federal Housing Finance Agency (FHFA) over the sale of mortgage related securities and paid CHF100 million to Britain under the terms of a tax deal between the two countries.

UBS is starting to bloom despite continued legal risks. (Keystone)

UBS will chalk up CHF865 million ($919 million) in legal bills, impairment of financial assets and “other significant provisions” in the second quarter. But the bank still expects to post a CHF690 million net profit, compared to CHF425 million in the corresponding period last year.

Zurich Cantonal Bank analyst Andres Venditti welcomed the news that UBS had finally dealt with the FHFA legal wrangle. UBS was one of 18 global banks to be sued for allegedly mis-selling mortgage backed securities during the financial crisis.

“UBS has paid a steep price, but the negotiations have been going on for some years so it is a positive step to finally get the litigation over with,” Venditti told swissinfo.ch. “Shareholders expect litigation costs to continue to be high in the next quarters. This may be a year of settlements, but the underlying outlook is positive.”

**Outstanding issues**

UBS was itself at pains to point out that its underlying business model can absorb hefty legal bills, announcing preliminary Q2 results a week earlier than its official financial presentation.

The bank surprised analysts who had forecast net profits of CHF558 million for the period of March to June. But the bank is still facing a number of outstanding lawsuits, not least as one of around 14 Swiss banks that may hand over confidential information to the US to clear up a long-standing tax evasion dispute.

The eventual cost of settling the ongoing row is not yet known, but it is not expected to come cheap. Last month, UBS was also fined €10 million by France for failing to comply with tax regulations.

**Julius Bär**

Switzerland's largest private bank, Julius Bär, announced the success of its takeover of Merrill Lynch's non-US wealth management operations as it posted its half year results on Monday.

Assets under management rose 15% to CHF218 billion during the last six months, Julius Bär said.

But the CHF99 million cost of absorbing the former Merrill Lynch business had an impact on results. The bank also raised the expected total integration costs from CHF400 million to around CHF455 million.

Julius Bär also paid out CHF28 million to the British tax authorities under the terms of a treaty with Switzerland to resolve tax dodging charges.

As a result of the one-off payments, net profits declined 30% from the same period last year to CHF114 million.

But the bank remained upbeat on Monday, highlighting the CHF12.8 billion in net new assets from wealthy individuals and families it had attracted in the last quarter. The bank also announced that its fire-sale of risky assets, a policy that was accelerated last autumn, had improved its robustness against future financial market shocks.

The second quarter has proved a prosperous one for large banking giants, with the US group of Goldman Sachs, JP Morgan, Citigroup, Morgan Stanley and Bank of America leading the way with forecast-beating results.

UBS rival Credit Suisse will report its Q2 results on Thursday, July 25.

Matthew Allen, swissinfo.ch

**READERS RECOMMEND** is a beta test, in cooperation with EPFL Artificial Intelligence Laboratory (LIA).

LIBOR RIGGING
Ex-UBS bosses accused of negligence, ignorance

INTEREST RATE SCANDAL
UBS fined for Libor rigging, traders charged

BENCHMARK FRAUD
Former UBS trader charged in Libor case

TAX EVASION
'Lax' UBS France hit by €10 million fine
France's banking regulator has fined Swiss bank UBS's French unit €10 million (CHF12.2 million) for “suspected illicit sales”, insufficiencies in controls and compliance with French clients. The bank says it will appeal. […]
Business   Banks in turmoil

LIBOR SCANDAL
Swiss banks face US Libor lawsuit
UBS and Credit Suisse are among 15 banks to be hit with a lawsuit by the United States government-owned mortgage finance company Freddie Mac as a result of the Libor manipulation scandal. […]
Business   Banks in turmoil   3 comments

US TAX DISPUTE
Cabinet approves bank data transfer solution
The Swiss government has agreed to give special permission for banks to cooperate with the United States authorities, paving the way for financial institutions to share data in a bid to avoid criminal charges for allegedly helping tax dodgers. […]
Business   Banks in turmoil

**Links**
UBS statement
UBS Switzerland

SWISSINFO.CH
Free mobile and TV apps allow you to access our highlights. You can also keep up with the latest Swiss news via a widget, or consult RSS feeds in your favorite reader.

RSS
Keep in touch

ANDROID
Android app with selected highlights
IPHONE & IPAD
iOS app with selected highlights
CONNECTED TV
Highlights for TV
WINDOWS 8

FACEBOOK
Curated content, by region
TWITTER
Global news feed
YOUTUBE
Video channel
GOOGLE+
Experimenting on G+

swissinfo.ch
International Service of the Swiss Broadcasting Corporation

© swissinfo.ch
**World partners:** Radio Netherlands Worldwide | Radio Sweden | Radio Canada International | NHK World | Radio Australia International

RSI RTR RTS SRF
swissinfo - a branch of the Swiss Broadcasting Corporation SRG SSR

**Figure 6: a swissinfo.ch's story of average length (2758 characters)**

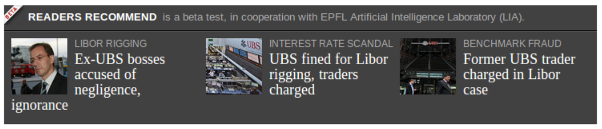

NEW
READERS RECOMMEND is a beta test, in cooperation with EPFL Artificial Intelligence Laboratory (LIA).
LIBOR RIGGING
Ex-UBS bosses accused of negligence, ignorance
INTEREST RATE SCANDAL
UBS fined for Libor rigging, traders charged
BENCHMARK FRAUD
Former UBS trader charged in Libor case

**Figure 7: swissinfo.ch's dynamic recommendation box**

Below the header, the *title* of the news story is displayed, then follows the *content* with one relevant image.

Recommendations generated by the PEN recsys framework are displayed in the *dynamic recommendations* box. The original design of the page limits to three recommended items due to aesthetic constraints (see Fig. 7). We believe that three recommendations restrain the user's choice. We do not know what is the optimal number of recommendations, but we plan to investigate this in the future.

Most of the news stories are very long and do not fit on the displayed area of the screen. swissinfo.ch has a *sticky menu* that does not disappear while scrolling down the page. It allows a quick navigation inside the current page. This menu contains a list of links pointing to parts of the news content, but also to the dynamic recommendations box (Fig. 7).

The author of a story can select related stories and place them in the *manual recommendations* box situated below the dynamic recommendations. We discuss the differences between manual and dynamic recommendations in the next Section.

Finally, users can submit feedbacks about the stories and they will appear just above the footer in the *users comments* area.

The layout of the page is extremely important. Since stories are rather long (2705 characters on average), the placement of recommendations on the page plays an important role in drawing users' attention. Because recommendations are at the bottom of the story, the increase in average clicks per visit (about 20%) is smaller than expected. Although there is a direct link in the sticky menu pointing to the recommendations, users need to scroll down the page to see the recommendation box, and we believe that users tend not to read the articles down to the end.

At the time of writing, recommendations are only available for stories and not tickers. It is planned to extend them to tickers as well. We will also experiment with different layouts by changing the position of the recommendation box either on the left, right or just below the title of the story.

## 6.4 Manual and Dynamic Recommendations

Manual recommendations are constructed by the author of the current news item. They take into account only the current article, but not the history and preferences of the reader. It is desirable to have an automatic way to create more personalized recommendations.

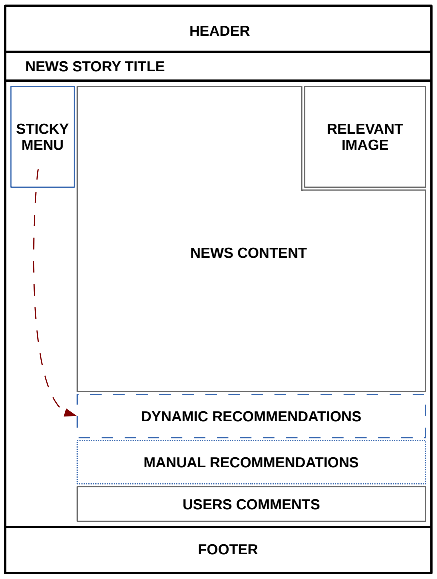


**Figure 8: swissinfo.ch's page layout. Dynamic (blue-dashed) and manual (blue-dotted) recommendations are at the bottom of the page.**

When generating dynamic recommendations, the system can take into account the set of manual recommendations in order to refine its recommendations.

We believe that manual recommendations can be replaced by a simple content-based recommender or a content retrieval system that do not rely on user's preferences. However, dynamic recommendations are more challenging to design and implement.

Manual recommendations might attract more users than personalized recommendations. Researchers should take into consideration the fact that one might drives more click than the other. However, it does not mean that the recommendations are bad, but that the interest of the users are different. To study this issue, it would be interesting to look at the click ratio of manual versus the click ratio of dynamic recommendations.

## 7. CONCLUSION AND FUTURE WORK

We presented the PEN recsys framework which aims at helping researchers and practitioners to conduct online evaluation of algorithms for news recommendations. The PEN recsys framework is fast, reliable, flexible and scalable. With the help of a simple control interface, it is possible to fine tune each recommender system and have a direct feedback of their performance.

Since July 2013, the PEN recsys framework is currently used by a news website for online evaluations of various recommender algorithms. At the time of writing, it is too early to draw any conclusion regarding performances of the different algorithms. However, we discussed issues that arose while conduction our online evaluation.

During our design, implementation and evaluation, a number of questions arose. We list them here as future direction of research, and we hope to answer some of them at the end of our online evaluation.

- Does a recommender algorithm behave the same in an offline setting than in a online setting?
- Which algorithm brings the best performance?
- What is the best tradeoff between fresh, popular and standard news items for the candidate set?
- How to make the recommendations more attractive to the users?
- What is the best placement for the recommendations?
- What is the optimal number of recommendations (3, 5, 10, ...)?
- What is the click ratio of manual vs dynamic recommendations?
- What is the optimal scheme to generate revenue?

## 8. REFERENCES

[1] Apache Software Foundation. Apache mahout. `http://mahout.apache.org`.

[2] D. Billsus and M. Pazzani. Adaptive news access. In *The adaptive web*, pages 550–570. Springer, 2007.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *J. of MLR*, 3:993–1022, 2003.

[4] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *Int. Conf. on World Wide Web*, pages 271–280, 2007.

[5] M. Ekstrand, M. Ludwig, J. Konstan, and J. Riedl. Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit. In *Int. Conf. on Recommender Systems*, pages 133–140, 2011.

[6] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: A free recommender system library. In *Int. Conf. on Recommender Systems*, pages 305–308, 2011.

[7] F. Garcin, C. Dimitrakakis, and B. Faltings. Personalized news recommendation with context trees. In *Int. Conf. on Recommender Systems*, 2013.

[8] F. Garcin, K. Zhou, B. Faltings, and V. Schickel. Personalized news recommendation based on collaborative filtering. In *Int. Conf. on Web Intelligence and Intelligent Agent Technology*, pages 437–441, 2012.

[9] M. Hoffman, F. Bach, and D. Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[10] E. Kirshenbaum, G. Forman, and M. Dugan. A live comparison of methods for personalized article recommendation at forbes.com. In *ECML/PKDD*, pages 51–66, 2012.

[11] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. Scene: a scalable two-stage personalized news recommendation system. In *SIGIR*, pages 125–134, 2011.

[12] P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer, 2011.

[13] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. Hellerstein. Graphlab: A new parallel framework for machine learning. In *Conf. on Uncertainty in Artificial Intelligence*, 2010.

[14] J. Picault, M. Ribière, D. Bonnefoy, and K. Mercer. How to get the recommender out of the lab? In *Recommender Systems Handbook*, pages 333–365. Springer, 2011.

[15] Research Studios Austria Forschungsgesellschaft mbH. easyrec. `http://www.easyrec.org`.

[16] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer, 2011.

[17] X. Su and T. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, pages 4:2–4:2, January 2009.

[18] M. Tavakolifard, J. Gulla, K. Almeroth, J. Ingvaldesn, G. Nygreen, and E. Berg. Tailored news in the palm of your hand: a multi-perspective transparent approach to news recommendation. In *Int. Conf. on World Wide Web*, pages 305–308, 2013.

[19] D. Vengroff. Reclab: a system for ecommerce recommender research with real data, context and feedback. In *Workshop on Context-awareness in Retrieval and Recommendation*, pages 31–38, 2011.