Devising News Recommendation Strategies with Process Mining Support

Elena Viorica Epure, Rebecca Deneckere, Camille Salinesi, Benjamin Kille, Jon Espen Ingvaldsen

Devising News Recommendation Strategies with Process Mining Support

Elena Epure, Rebecca Deneckere, Camille Salinesi
CRI, Panthéon-Sorbonne
Paris, France
Elena.Epure@malix.univ-paris1.fr
Rebecca.Deneckere, Camille.Salinesi@univ-paris1.fr

Benjamin Kille
DAI-Lab, TU Berlin
Berlin, Germany
benjamin.kille@dai-labor.de

Jon Espen Ingvaldsen
IDI, NTNU
Trondheim, Norway
jonespi@idi.ntnu.no

***Abstract*—News media is in a digital transformation, disrupting their existing business models. Many news media houses are looking into recommender systems as a part of their digital strategies. However, the social role of journalism, existing publishing platforms and news as a continuous data stream infer particular challenges for applying standard recommender technologies. This paper explores how news recommendation can go beyond popularity and recency and take advantage of content quality metrics and interaction patterns. This knowledge is derived through adapting process mining for usage with web logs. The proposal is evaluated on real event logs from a German news publisher, revealing encouraging results.**

I. Introduction

The digital transformation has reshaped news media's ecosystem, conveying the adoption of web portals, mobile solutions, and recommendation technologies as part of their strategies. From a business perspective, news media has also been facing a higher competition in engaging their readers for maintaining advertising revenues. With data-driven technologies, news publishers have managed to target their content distribution to individual readers' interests.

However, devising an effective news recommendation strategy is still challenging because of various reasons: the very high rate at which new items are published (data sparsity); users' relatively unstable preferences influenced by breaking news and such; potentially unreliable implicit user feedback. These challenges are amplified for smaller news providers, where technical resources and knowledge are more limited [7]. Thus, grasping users' interests and their reading behavior is necessary for publishers to create a strategy addressing these challenges.

In this paper, we propose and adapt *process mining* [1], a suite of data mining techniques initially applied in the context of Business Informatics, to reveal visual flow models describing news reading behavior. In particular, we analyze users' interaction with news categories. The discovered knowledge drives the implementation of a hybrid recommendation strategy. The solution is evaluated in a near-to-online setup with a Berlin-based newspaper, which accommodates about 3 million visits per week on average. Although web logs have their limitations, we show that we are able to reveal valuable insights about how news categories are consumed and how they relate to each other. Our proposed solution improves the recommendations over a popularity and recency-focused baseline in almost 90% of the cases. It also diversifies the suggestions.

II. Process Mining for News Recommendation Strategy

A. Process Mining Presentation

Process mining consists of a suite of techniques and tools, which has been effectively used to discover and visually represent complex process models from logged traces of information systems' users [1] and, more recently, of web users [15]. A process model can be formalized by a quadruple, $P = (S_{\text{start}}, S_{\text{stop}}, A, T)$, where each element is:

- $A = \{a_n\}_{n=1}^{N}$ the finite set of process activities,
- $S_{\text{start}} \neq \emptyset$ the set of initial process states,
- $S_{\text{stop}} \neq \emptyset$ the set of final process states,
- $T \subseteq ((S_{\text{start}} \times A) \bigcup (A \times A) \bigcup (A \times S_{\text{stop}}))$ the finite set of transitions between activities, or activities and states.

Further, using the transition notation, several types of relations could be identified:

- *Sequence*: $t \in T : t = a_i \prec a_j$ or $t = s_{\text{start}} \prec a_k$ or $t = a_l \prec s_{\text{stop}}$ with $s_{\text{start}} \in S_{\text{start}}$, $s_{\text{stop}} \in S_{\text{stop}}$, $a_i, a_j, a_k, a_l \in A$ and $i \neq j$. The interpretation of $a_i \prec a_j$ is that once the activity a_i is finished, the activity a_j follows. The transitions $s_{\text{start}} \prec a_k$ and $a_l \prec s_{\text{stop}}$ are implicit transitions marking the start and the end of the process.
- *Loop*: $t \in T : t = a_i \prec a_i$ where $a_i \in A$. The interpretation is that once the activity a_i is finished, it starts again.
- *XOR-split*: $t_m, t_n \in T, t_m = a_i \prec a_j, t_n = a_i \prec a_k$ and $t_m \oplus t_n$ where $a_i, a_j, a_k \in A$ and $j \neq k$. The interpretation is that either t_m or t_n takes place; once the activity a_i is finished, either the activity a_j or a_k follows.

More relations exist but only the presented ones are within the scope of this work. A process execution $E = s_{\text{start}} \prec \ldots \prec a_i \prec \ldots \prec s_{\text{stop}}$ represents a sequence in the process from one start state to one end state, with at least one activity in between. A process can contain multiple execution paths given the presence of loop and XOR-split relations.

In most of software application nowadays, either web-based or desktop-based, events are generated and logged during their usage. Process mining exploits these events for revealing process models in a bottom-up approach. Let

e be such an event captured during the user interaction with an application. The minimum information about an event required by a process mining technique is: e_{id} the event's unique identifier, e_t the event's timestamp, $e_a \in A$ the event's associated activity. The activity is usually inferred and attached to the event in a pre-processing phase, preceding the mining phase. This happens by reasoning on the existing event's meta-data. Example of activities include "access account", "add new customer", and "delete registration".

A specific process execution E by a user $u \in U$ is logged as a trace $\tau = \{e_1, e_2, \ldots, e_r\}$. Let $L = \{\tau_v\}_{v=1}^{V}$ be a log containing a finite set of traces. The process mining technique is a function γ that maps the log L on a process model P. For one specific log, multiple definitions of the function γ could be proposed as different algorithms. Also, variations of the same function γ could exist by parameter manipulation [1]. Often, these parameters can be changed interactively in the interface of the process mining tool.

Apart from discovering formalized process models, an effective representation of these models for enabling decision support is central to process mining. This principle represents in fact one major difference to data mining. A particular challenge that process mining must deal with is the "spaghetti processes'", the type of processes that are very complex and highly unstructured, resulting in unreadable graphical representations. For this, several techniques have been aimed not only at discovering the function γ but also at proposing ways of simplifying the visualization while keeping a fit representation of the log L. An example of algorithm in this direction is the Fuzzy Miner [9], which follows an approach inspired from cartography. It mines processes and represents them in readable, visual models by leveraging the level of representative behavior through abstraction, emphasis, customization and aggregation. The level of details regarding a process is interactively set by the user using the interface of the process mining tool [9].

To ensure effective decision making and knowledge extraction support within *news organizations*, data exploration should be visual, interactive and intuitive [10]. This is why process mining is explored for devising recommendation strategies. Also, encouraging results have been obtained by recent works in web mining using process mining [3], [15].

B. Log Preparation for Mining News Reading Processes

Let's consider a user u that visits the web news portal and navigates to another page to read the article art by clicking its title advertised on the screen. In that moment, a click event, e^c is generated for capturing the fact that a recommendation succeeded as the advertised news article was clicked. Further, once the requested article is rendered and shown in browser, an impression event, e^i, is generated and stored with the following meta-data: the user's id u_{id} (the session's id), the article's id art_{id}, the article's text including the title and abstract art_{text}, the set of categories associated with the article $art_{cat} = \{c_m\}_{m=1}^{M}$, the timestamp θ, and other features collected by the tracking component such as the type of browser, operating system, etc. Each article could have multiple categories associated. In order to generate the event logs necessary for process mining, the following mapping are made:

- The session id (cookie-based) is associated with the unique identifier of the process execution's trace;
- For each impression event, the categories of the read article are sorted, concatenated by commas, and associated with the activity of one trace event;
- For each impression event, its timestamp is associated with the timestamp of one trace event;
- For each impression event, a unique identifier is generated during the pre-processing and associated with the id of one trace event.

The process activities are mapped on articles' categories because, in the literature, users' news reading interests are often represented by categories [5], [12], [13]; hence, we associate the news reading behavior to a process of transitioning between news interests. In addition, this overcomes the data sparsity issue, especially prevalent in the news domain. The format chosen for the log files is "csv", character-separated values. Table I shows an example of processed web log events from news reading.

TABLE I
WEB LOG EXAMPLE; THE EVENTS' IDS ARE OMITTED.

Trace Id	Timestamp	Activity
64183	2015/09/28 09:01:20	Health Sport Wellness..
64183	2015/09/28 09:01:33	Travel Tourism Navigation
49616	2015/09/28 09:07:02	Banks Finance Insurance

C. Process Mining Tools for News Reading Processes

Two process mining tools are reported to be extensively used by related works: ProM [16] and Disco [8]. ProM contains a more extensive variety of mining algorithms but is less suited for handling very big event logs. By contrary, Disco is a commercial tool that can handle big data. It is based mainly on the Fuzzy Miner technique [9]. Considering these, we opt to use Disco in the current work as web logs from news reading contain millions of events per month.

Examples of mined models are presented in Figure 1. Each rectangle in Figure 1(b) represents an activity. We notice in Figure 1(b) the default states, start and stop, represented as circles, at the top and bottom of the model. The arrows mark the transitions from one activity to another. The size of the arrow and the color of activities give information about metrics. For example, the frequency of a transition is marked by the thickness of the arrow's line while the frequency of an activity is revealed by the color's intensity (darker, more frequent). Figure 1 also shows the explorative feature of process mining tools to maneuver from spaghettiness and noise to valuable flow relations and actionable insights. Figure 1 (a) shows how newspaper's event logs produce a complex "spaghetti" model when we visualize all activities

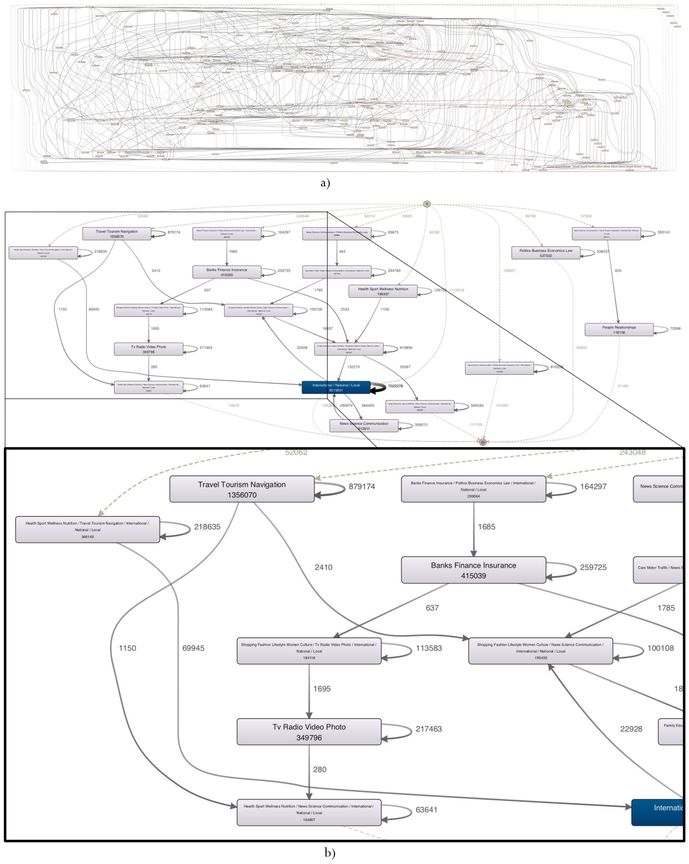


Fig. 1. (a) Example of mined "spaghetti" model. (b) The same model showing the 10 % most frequent activities and 5 % most representative behavior. The model is zoomed in around the "Travel Tourism Navigation"

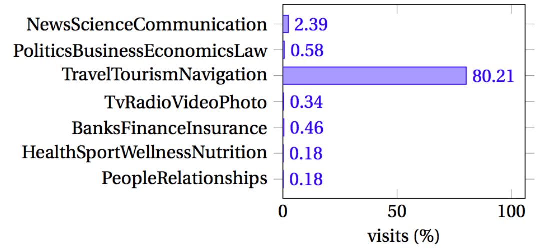


Fig. 2. Proportion of cases when common news categories are accessed after readings of a "Travel Tourism Navigation"-article

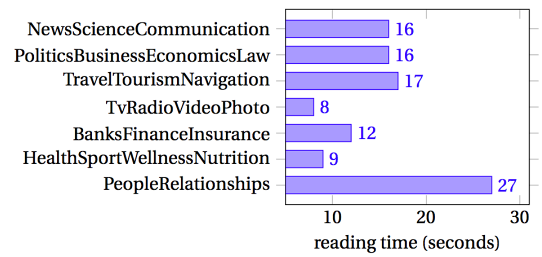


Fig. 3. Average reading times for a subset of common news categories accessed after readings of a "Travel Tourism Navigation"-article

and transitions. Figure 1 (b) shows a model extracted from the same event log as in (a), but here frequency thresholds on activities and transitions are applied to create a simpler and more understandable model.

Process mining tools provide process models enriched with various performance metrics. Figure 1 presents the activities and transitions enriched with frequency values (the numbers associated with the arrows and rectangles). Process mining also describes temporal aspects of the process flows such as: how much time an activity lasts on average, what is the average, minimum or maximum time between two activities [11]. Disco, the process mining tool, creates a view on the same model emphasizing on the temporal metrics. In our knowledge extraction phase, we often switched between the two perspectives of the same models with the help of the tool. Moreover, the tool provides these perspective for individual sessions, thus enabling us to make comparisons. This work omits such comparisons due to space limitations.

D. Revealing News Reading Processes and Related Knowledge

In the reading news processes, the frequency values, regarding activities and transitions, are a measure of popularity of standalone news categories and of relations between news categories. In Figure 1(b), we can see the news category "Travel Tourism Navigation" is accessed 1356070 times. Readings of this news category is also observed to be followed by readings within the same category 879174 times (the loop transition). Mined models reveal that loops are common for all news categories. Figure 4 shows how many categories users read during both short and long reading sessions, for Berlin newspaper in September 2014. Here we can see that most short and long reading sessions contain a single category. This reveals that the next read article is likely to be within the same category as the previous article. When we look at the alternative paths from start to end in this model, we also see how different category-activities cluster and form sequential relationships. In Figure 1 (b), "Travel Tourism Navigation" has two outgoing transitions to other category-activities at the selected filtering levels. This is actionable insight that could be exploited in the definition of recommendation strategies.

Figure 2 shows some of the most common news categories following after readings of articles from category "Travel Tourism Navigation". In 80 % of the cases, the next article is related to the same category. The second most popular category, following "Travel Tourism Navigation", is "News Science Communication" with 2 % of the traffic. Although these frequency values can predict what the user will click next, they do not describe to which extent the users actually appreciate or engage with the content. When we manually analyzed the process models, we noticed that the frequency of transitions sometimes delivered insufficient information.

The timestamp differences between two events in a

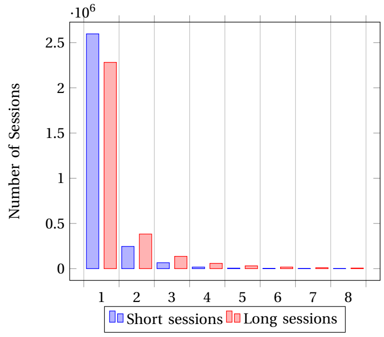


Fig. 4. Number of sessions vs. number of unique categories per session.

session can tell us how long the user spent reading a specific article. If, for instance, the time between two subsequent events with the categories "Sport" and "Economics" is short, this means that users did not read the sport-related article but they quickly navigated to the economics-related article. Contrarily, if the timestamp difference is too long, this is a signal that the user has been inactive and that the "Economics" reading is actually the first event in a new reading episode. To compare the alignment of popularity and reading time, we extracted and analyzed reading episodes with a maximum 5 minutes delay between events. We traced the sequential reading events in this dataset and calculated the average reading time for each of the next news categories observed. Figure 3 shows the average reading time on a subset of common categories following after a "Travel Tourism Navigation" article. We can see that the category "People Relationships" has the highest value with 27 s. When we compare the popularity and reading time statistics, we see that news articles of this category are only accessed 0.18 % of the cases, but when they are accessed users tend to appreciate and engage with the content.

In summary, the analysis driven by process mining revealed multiple insights about the news reading processes:

- Loops: news article readings are likely to be followed by readings within the same news category.
- Category relations: the process models reveal strong sequences and XOR-splits of specific news categories.
- Reading time: there could be an inconsistency between frequently clicked categories and the time users spent engaging with those types of articles.
- Loyalty: Users are loyal to few news categories in both short and long reading sessions.

III. News Recommendation Strategies

A. Baseline Strategy

A baseline strategy takes into account the recency and popularity of news articles. The *recency* refers to the property of an article to have been recently read by the crowd. The *popularity* refers to the number of times an articles has been read. The popularity and recency can be captured through a ring buffer implementation. We define a list L, which can take elements up to a fixed number k and the index i pointing to the next entry to be populated in L. Whenever the system observes a visitor reading a news article art, it adds the article to the list at the position i and moves the index to the next entry (if i is currently at position k, it is reset to 1 creating a circular pattern). In this way, the collective reading preferences are captured. Note that duplicates are allowed.

At recommendation, the system takes the current list and iterates backwards from the current index. The system returns the first article art_j (or first N articles) that mismatches the article that the visitor is currently reading art_i, $i \neq j$. Consequently, the baseline guarantees that visitors are likely to receive the most recent news. In addition, the probability of an article to be suggested is proportional to the article's popularity. This is ensured by allowing duplicates in the ring buffer. The baseline has proved to help the system to follow trends and distribute important news [14].

B. Process-driven Strategy

Based on related literature, past users' reading interests, expressed as news categories, could affect the reading choices apart from popularity and recency. However, the relations between categories, thus the process over the reading interests, has been less investigated. As shown, the baseline strategy covers popularity and recency, but it does not consider the users' personal reading interests and habits to transition between those interests in reading. Process mining provides insights to narrow this gap. Process mining delivered a set of patterns describing readers' transitioning between categories of news and other knowledge valuable in data pre-processing such as selecting just transitions reflecting engaged reading. Let the set of transitions from a specific process model P be: $\mathcal{T} = \{(c_s, c_d, p_{sd}) | \exists c_s, c_d, (c_s \prec c_d) \in P\}$. Therein, we denote the individual category or a combination of categories by c. We consider p_{sd} the degree of expected belief with which the transition to c_d from c_s is observed. In order to maintain the popularity and recency constraints in recommendation, a separate ring buffer for each category c is created to store read articles from this category, similarly to the baseline version. Subsequently, the system has to determine to which ring buffer to delegate the recommendation request. We derive this information from the p_{sd} values and consider the maximum degree of belief, given the source category c_s.

The set of categories comprises relatively few elements compared to the immense number of articles. Nonetheless, editors struggle to assign a single category to a given article especially for those articles covering topics related to multiple domains. For instance, imagine an article about the economic implications of a major sports event. As a result, a majority of articles has multiple category labels assigned. This situation presents some challenges for the implementation of the proposed recommendation strategy. We could approach this by creating a ring buffer for all combinations of categories. However, even with few existing categories, this approach would require a large collection of ring buffers. For instance, 8 initial categories would result in 256 ring buffers needed (exponentially growing). Another problem with this solution is the fact that some final categories would contain very few or even a single article. Thus, the system would always suggest the same article in some cases.

We opt for another approach in order to circumvent these issues. We iterate over the collection of transitions $\mathcal{T}$ extracting all combinations of categories for each transition. This step is costly, but we pre-compute the links between categories only once in advance. Thus, repetition is avoided and the complexity at run-time remains unaffected. Having extracted all possible transitions, we aggregate p_{sd} for all combinations of categories c_s, c_d. This resulted in a symmetric matrix $\mathcal{M}$, where the rows and columns correspond to the categories, and the values to the probabilities of each transition. Note that each row, respectively column, refers to a single news category here. Whenever the system receive a request, we decode the contained categories to a binary vector. This vector has 1 for each present category and 0 for each absent category. The category whose ring buffer to use for suggesting a recommendation is determined by multiplying the request vector with $\mathcal{M}$. Finally, the category corresponding to the maximum value in the resulted vector is chosen. If maximum values coincide for multiple categories, a category is selected at random.

IV. News Recommendation Evaluation

A. Evaluation Strategy and Metrics

In order to evaluate the performance of a recommendation strategy, two types of evaluations are possible: online and near-to-online. The *online* or *live* evaluation uses as performance metric the click-through rate (CTR). This metric calculates the proportion of click events to the total number of suggested articles. The *near-to-online* evaluation verifies how accurately a recommender algorithm predicts the articles interesting for the user. An article is interesting for the user if the user will read it within a short time frame. The recommender algorithm receives the sequence of impression events and the sequence of click events. For each impression event e^i, the recommender algorithm provides a list of 4 recommendations $R = \{art_{r1}, art_{r2}, art_{r3}, art_{r4}\}$; 4 corresponds to the required number of recommendations for the newspaper under study.

Subsequently, in the evaluation, it is checked if any of the suggested articles appears among the future impression events. A score of 1 is assigned to the requests if a match is detected and 0 otherwise. Note that this procedure differs from click-rate evaluation. The user is not actually required to click on the recommendation advertised on the news portal's web page. Instead, it is checked whether the user requests the recommended article in the session, the session being generally very short. We define a session, $S = \{art_0, \ldots, art_N\}$ as sequences of articles read corresponding to the impression events $\{e^i_0, \ldots, e^i_N\}$ and the user u. The scoring function is defined:

$$\xi(art_i, R) = \begin{cases} 1 & \text{if } \exists art_r \in R \text{ such that } art_r \in S_{>i} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We denote all readings that occur after art_i as $S_{>i}$. Eventually, we obtain a normalized score for each session (2), which is bound to the closed interval $[0, 1]$.

$$Q(S) = \frac{1}{|S|} \sum_{art_i \in S} \xi(art_i, R) \tag{2}$$

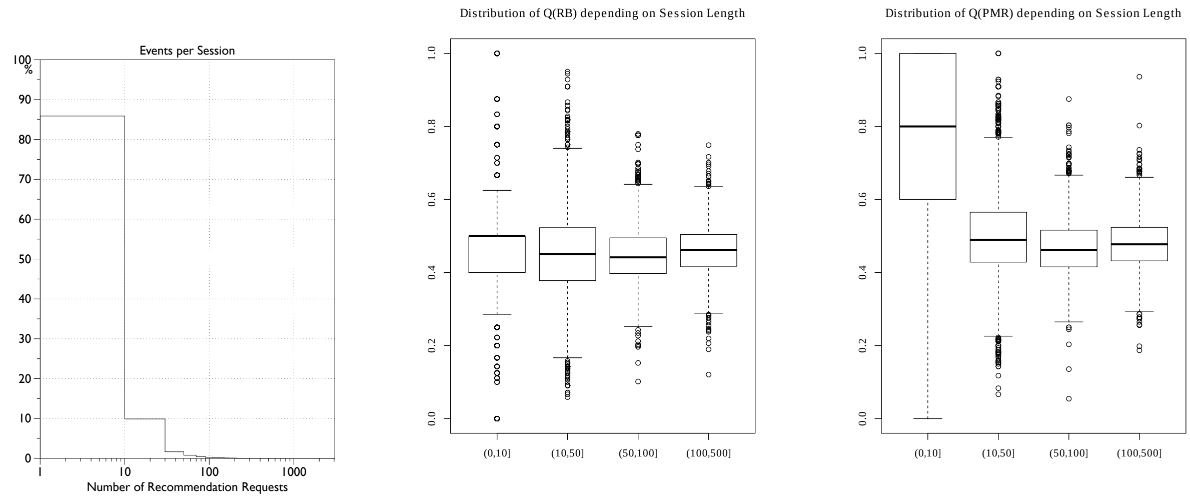
B. Evaluation Objective and Data

The main objective is to evaluate and compare the proposed recommendation strategies in a near-to-online setup, using the metric previously introduced. We refer to the baseline as *BaseRecmm* and to the recommender enhanced with process mining knowledge as *ProcessRecmm*. Previous evaluations have shown that BaseRecmm yields results comparable to other more complex recommender algorithms [14]. We used a data set with ≈ 1.8 million recommendation requests in 224 109 sessions from February 2015. The data set spanned ≈ 4 weeks. Both BaseRecmm and ProcessRecmm provided 4 recommendations for each request. This corresponds to the number of slots available on the publisher's news portal.

The past news reading behavior knowledge was extracted with process mining from September 2014. The rational for considering only one month was influenced by the capacity of the process mining tool. Being desktop-based, we had a memory constraint. The news publisher's monthly data amounts for 20 GB to 30 GB excluding articles' texts. After we extracted the information relevant for process mining, we were able to shrink the data to 4 GB to 5 GB, which allowed us to work on a 8 GB RAM computer for the data analysis with the process mining tool. September 2014 was chosen in particular because similar to February 2015 starts also on a Monday and the distance between them ensures that completely different users are concerned.

C. Results and Discussion

First, we analyzed to what degree both recommendation strategies provide identical suggestions. Initially, the algorithms fail to provide the requested number of suggestions. As soon as they collected enough information, they managed to respond with the designated number of items. Both systems provided 4 items in more than 99.9 % of the sessions used in the evaluation. Further, for each request, we determined how many common items have been recommended by both algorithms. In 50.2 % of the requests, all 4 items were identical. In 30.1 % of the cases, ProcessRecmm and BaseRecmm agreed on only 3 items. The algorithms delivered two identical items in 7.0 % cases. A single item matched 8.3 % of the time. Both recommenders completely disagreed in 4.3 % of requests. This showed that ProcessRecmm and BaseRecmm ensured differently the diversification of recommendations.



(a) Histogram events per session. >85 % of sessions have max 10 events.

(b) Boxplot BaseRecmm. Score relatively constant independent of session length.

(c) Boxplot ProcessRecmm. Scores much better than BaseRecmm for short sessions.

Fig. 5. Analysis of the proposed recommendation strategies: baseline and driven by news reading processes.

Then, we measured the quality of recommendations in terms of successful predictions. Both ProcessRecmm and BaseRecmm received each article in a sequence of readings and provided a list of 4 recommendations. Subsequently, we checked whether any of the suggested articles appeared as future reading in the same session. The sessions used in the evaluation differed in length. Figure 5(a) shows the proportion of sessions with a given number of events. A majority of sessions exhibited at most 10 events. We compared the scores of ProcessRecmm and BaseRecmm for different groups of sessions. Figure 5 illustrates our findings. The baseline scored comparably for sessions of varying lengths as Figure 5(b) shows. In contrast, ProcessRecmm scored noticeably higher for short sessions as Figure 5(c) confirms. When analyzing in detail, in 88.8 % of the cases, ProcessRecmm's score exceeded the baseline.

In summary, the recommendation strategy incorporating knowledge on news reading processes improved the recommendations significantly compared to a popularity and recency-based strategy and diversified more the suggestions. A more extensive analysis is required, especially with data from different months for both prediction and training.

V. Related Work

Billsus and Pazzani [4] describe various ways for visitors to receive personalized news. Their taxonomy includes news content personalization, adaptive news navigation, contextual access, and news aggregation. Das *et al.* [5] illustrate the news recommender system deployed at Google. Their solution exploits the transitions between articles merely through a covisitation metric; that stands for a popularity mechanism too. Existing works based on processes exist. Li *et al.* [12] consider the process activities being users and articles. Ahmed *et al.* [2] tackled personalized news recommendation via a transition graph with three types of activities: views, clicks and documents. Compared to these, our approach is based on transitions between news categories. Liu *et al.* [13] introduced a follow-up work on Google News. Their frameworks model visitors' current interests as news categories. It is the closest to our work. Although, the category prediction is based on Bayesian networks and the recommendation of articles is different. Doychev *et al.* [6] describe several approaches for news recommendation algorithms evaluated on CLEF-NewsREEL 2014 competition dataset. Their data analysis also shows that users tend to read news from the same category and a few dominant ones account for the majority of items that are read. They also found valuable patterns of activity between certain less popular categories and point out attempts to exploit these in future work.

VI. Conclusions

In this paper, we extract knowledge from news portals' logged data as *dynamic models that capture the users' reading interests and behavior* using process mining. We further, integrated this knowledge into *a recommendation strategy* together with articles' recency and popularity aspects. The proposed solution proves encouraging in an evaluation with one month worth of real data from a German news publisher. Further work aims at expanding the experiments for results' generalization. Compared to related works, we show how to devise a low-level recommendation strategy by being supported by a high-level, visual analysis. This could empower news organizations to conduct such analyses in-house and draw informed strategy's decisions without a thorough level of technical knowledge.

References

[1] W. v. d. Aalst. *Process mining.* Springer, 2011.

[2] A. Ahmed, C. H. Teo, S. V. N. Vishwanathan, and A. Smola. Fair and Balanced : Learning to Present News Stories. In *WSDM*, pages 333–342, 2012.

[3] H. Awatef, B. Gueni, M. Fhima, and S. David. Process mining in the education domain. *International Journal on Advances in Intelligent Systems*, 8(1-2), 2015.

[4] D. Billsus and M. J. Pazzani. Adaptive News Access. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, chapter 18, pages 550–570. Springer, 2007.

[5] A. Das, M. Datar, A. Garg, and S. Rajaram. Google News Personalization : Scalable Online. In *WWW*, pages 271–280, 2007.

[6] D. Doychev, A. Lawlor, R. Rafter, and B. Smyth. An analysis of recommender algorithms for online news. In *CLEF (Working Notes)*, pages 825–836, 2014.

[7] E. V. Epure, J. E. Ingvaldsen, R. Deneckere, and C. Salinesi. Process mining for recommender strategies support in news media. In *2016 IEEE 10th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, June 2016.

[8] C. W. Gunther and A. Rozinat. Disco: Discover your processes. In *BPM*, 2012.

[9] C. W. Gunther and W. M. P. Van Der Aalst. Fuzzy mining: Adaptive process simplification based on multi-perspective metrics. In *Proceedings of the 5th International Conference on Business Process Management*, BPM'07, pages 328–343, Berlin, Heidelberg, 2007. Springer-Verlag.

[10] T. C.-K. Huang, C.-C. Liu, and D.-C. Chang. An empirical investigation of factors influencing the adoption of data mining tools. *International Journal of Information Management*, 32(3):257 – 270, 2012.

[11] J. E. Ingvaldsen and J. A. Gulla. Industrial application of semantic process mining. *Enterprise Information Systems*, 6(2):139–163, 2012.

[12] L. Li and T. Li. News Recommendation via Hypergraph Learning : Encapsulation of User Behavior and News Content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 305–314, 2013.

[13] J. Liu, P. Dolan, and E. R. Pedersen. Personalized News Recommendation Based on Click Behavior. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 31–40, 2010.

[14] A. Lommatzsch. *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, chapter Real-Time News Recommendation Using Context-Aware Ensembles, pages 51–62. Springer International Publishing, Cham, 2014.

[15] N. Poggi, V. Muthusamy, D. Carrera, and R. Khalaf. *Business Process Management: 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*, chapter Business Process Mining from E-Commerce Web Logs, pages 65–80. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[16] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst. The prom framework: A new era in process mining tool support. In *Proceedings of the 26th International Conference on Applications and Theory of Petri Nets*, ICATPN'05, pages 444–454, Berlin, Heidelberg, 2005. Springer-Verlag.