January 2023

DEEPSORTING: CUSTOMER-CENTRIC SEQUENCE-TO-SEQUENCE NOTIFICATION PRIORITIZATION

Pengfei Sun

Qihong Shao

Derek W. Engi

DEEPSORTING: CUSTOMER-CENTRIC SEQUENCE-TO-SEQUENCE NOTIFICATION PRIORITIZATION

AUTHORS:
Pengfei Sun
Qihong Shao
Derek W Engi

ABSTRACT

Network operators may control many thousands of devices and, as a consequence, they may be bombarded with notifications (e.g., posture assessments and exceptions for devices that do not comply with standards or which require remediation actions to circumvent security issues) and can be discouraged by unimportant or irrelevant information. To address the challenge that was described above, techniques are presented herein that support a process for intelligently filtering and prioritizing notifications to reduce noise and deliver recommendations that will provide the greatest impact to an environment. Aspects of the presented techniques encompass a smart recommendation notification system that prioritizes network actions based on multiple embedding spaces and dimensions; the use of business and financial logic, a persona, and a network operating state for reducing network actions into a prioritized output; the use of click-through and sequence mining to establish a ground truth of event prioritization of a network operator; an adaptive learning method for tracking proposed network recommendations to the final action that may be executed by a network operator; and a method for reducing multi-step recommendations based on an identification of the most efficient sequence of events based on a network operator implementation.

DETAILED DESCRIPTION

Network operators may control thousands and even tens of thousands of devices as they accomplish their business and solution goals. As new features are developed, as industry best practices are adopted, and as security vulnerabilities are discovered, network operators are bombarded with posture assessments and exceptions for devices that do not comply with standards or which require remediation actions to circumvent security issues. Each of those individual actions may be referred to herein as a “notification,” indicating that a remediation step must be taken on a device to increase the overall health of a network device. Operators face an overwhelming number of such notifications and can be discouraged by unimportant or irrelevant information.

To address the challenge that was described above, techniques are presented herein that support a process for intelligently filtering and prioritizing notifications that may be issued to a customer to reduce noise and deliver recommendations that will have the greatest impact to an environment. Figure 1, below, depicts elements of an exemplary arrangement that is possible according to the techniques presented herein and which is reflective of the above discussion.

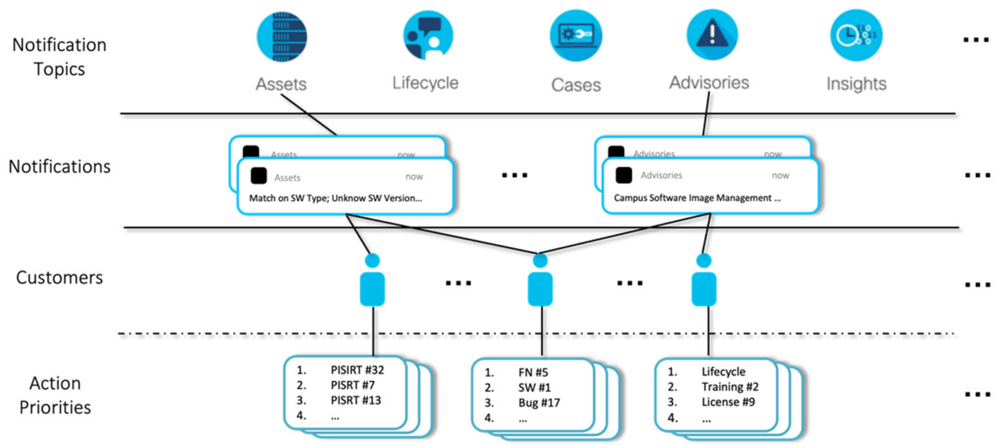

Figure 1: Exemplary Notification System and Customer Actions

The techniques presented herein encompass an intelligent filtering system that generates a prioritized email sequence based on the risk factors that are associated with a given customer's business or network. It is important to note that while email is used as an example in the instant narrative, according to aspects of the presented techniques the output of the depicted model may be consumed in any number of ways.

To provide an efficient and customized smart notification system that can automatically and accurately prioritize many incoming notifications, five main challenges must be addressed when sorting a notification list that best reflects a customer's interests. According to those challenges, a notification system must (1) prioritize recommendations based on multiple factors (including industry particulars, customer specific logic, and inferred network logic); (2) must be the least tolerant of mis-prioritizing urgent and financially significant recommendations; (3) must directly affect the potential action priority, and therefore the system should minimize the potential monetary cost and time cost for a prioritized action; (4) must be able to minimize the priorities of irrelevant or non-urgent recommendations; and (5) must be robust and resilient to emergencies and be adaptive and evolve with emerging threats and best practices.

To enhance a customer's experience and to help promote the most urgent tasks, according to the techniques presented herein a DeepSorting model may utilize a combination of multiple input features ranging from generalizable business metrics to highly customized and specific network metrics. Such multi-perspective data sources may be used to build both global and customer-specific models.

Figure 2, below, presents elements of an exemplary arrangement (including a DeepSorting capability) that is possible according to the techniques presented herein and which is reflective of the above discussion.

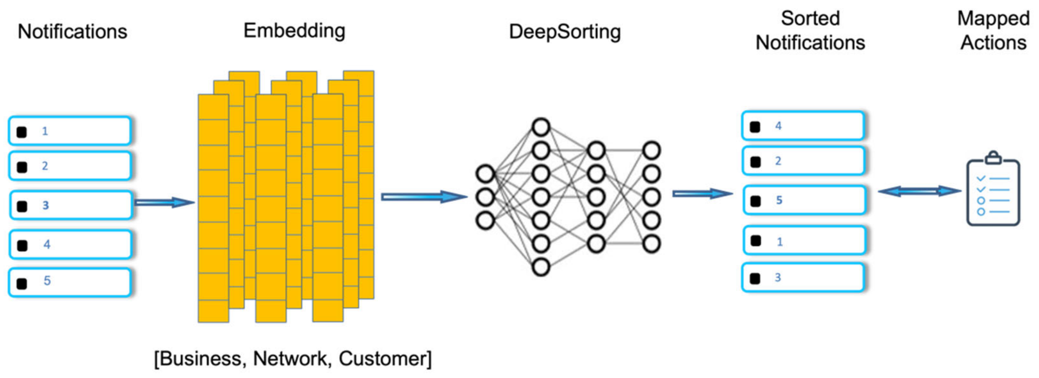

Figure 2: DeepSorting for Prioritization of Notifications

As depicted in Figure 2, above, notifications or posture assessments (along with their related context information) may be parsed and converted into domain embeddings, in which multiple generative models regarding business, network, and customer factors may be employed. The generated domain and context embeddings may then be fed into a DeepSorting model, where the predicted notification sequences will reflect the relative prioritizations through the rearranged positions. Further, the sorted notification sequences may be applied to generate action lists for customers.

To be more robust regarding newly emerged notifications that were not included in a previous learning pool, a continuous learning module may incorporate the samples with a large discrepancy between predicted action and customer behavior. Figure 3, below, depicts elements of such an approach.

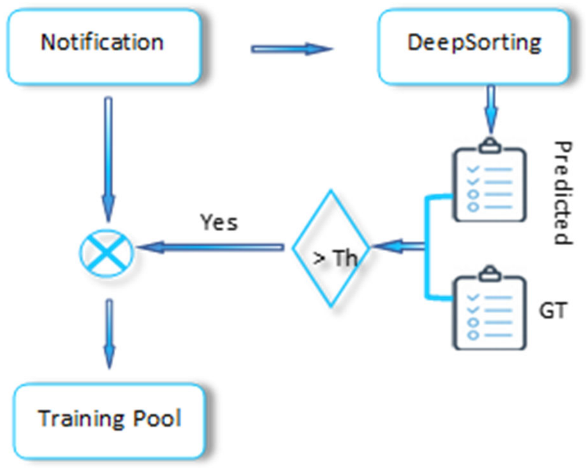

Figure 3: Continuous Training Flow

As shown in Figure 3, above, once the discrepancy between a predicted action list and a ground truth (GT) is greater than a threshold value (Th), an incoming notification may be automatically added to a training pool to prepare for the next update of a DeepSorting model.

The next portion of the instant narrative describes and illustrates one system architecture that is possible according to the techniques presented herein.

By extracting the global business features and localized customer and network context features from notification sequences, a DeepSorting system, according to the techniques presented herein, may utilize the sequence-to-sequence (Seq2Seq) family of machine learning approaches to translate a notification batch into sorted notification sequences. Further, the sorted notification sequences may be summarized into prioritized actionable insights. Figure 4, below, depicts elements of such an approach.

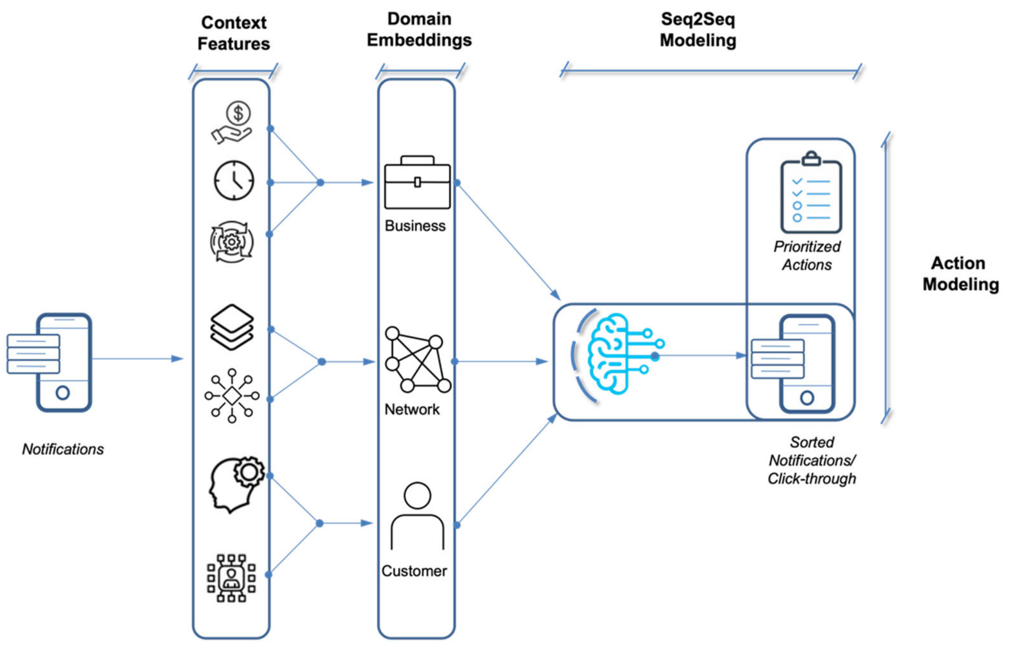

Figure 4: Exemplary DeepSorting System Architecture

Within the system architecture that is shown in Figure 4, above, after the input raw notifications are decomposed and parsed categorically, the notification sequences may be transformed into a context feature space. The categorized context features may then be fed into the different generative models to output domain embeddings.

According to the techniques presented herein, during a training phase a DeepSorting model may utilize the domain embeddings as an input whereby a customer's responses (e.g., a click-through, repeated open and click rates, etc.) may be applied as the GT of the priorities. Due to the advantage of the intrinsic characteristics of a Seq2Seq model (which will be described below), the customer's responses do not necessarily need to exactly match the input notification sequences, thus automatically allowing more flexibility in the model training.

Further, during an inferring phase the DeepSorting model may predict the prioritized notification sequences and the sorted notification sequences may be assumed to be the approximation of the customers' responses. However, if a large discrepancy exists between the GT and the predicted customer's responses, any outliers may be added to the training data pool for future model updating.

During the training phase, the sorted notifications may be represented by a customer's historical click-through data on previous notification data. In this way, the click-through data may serve as the GT to minimize the entropy loss of sorted notifications.

Since the open and click rates may not be sufficient to always reflect priorities for the notifications, aspects of the techniques presented herein may employ a comprehensive scoring system as the DeepSorting loss function. Such a scoring system may not only count open and click rates but also other factors such as repeated open and click, a first response time, and the time spent on each open and click.

It is important to note that a customer's eventual remediation action may be different from their click-through behavior. The underlying rational is that a click-through relies on limited information while a final action will be determined by the full released knowledge. To emphasize this discrepancy in the instant DeepSorting system, an action model may be employed to map a customer's click-through behaviors to their final actions. In this way, a DeepSorting system may provide recommendations according to a customer's general click-through behaviors. From this perspective, a DeepSorting system is an intelligent system that can approximate a domain expert's comprehensive evaluations and recommendations and track the relationship between initial click-through behavior and final remediation actions.

The next portion of the instant narrative describes and illustrates elements of a context features and domain embeddings capability that is possible according to the techniques presented herein.

Within a DeepSorting system, according to the techniques presented herein, the input notifications and the context information firstly may be parsed into different categories such as business, customer, and network. As shown in Figures 5a through 5c, below, the context features in each category may be extracted based on the predefined topics that reflect the notification's impact from a global level to localized scenarios.

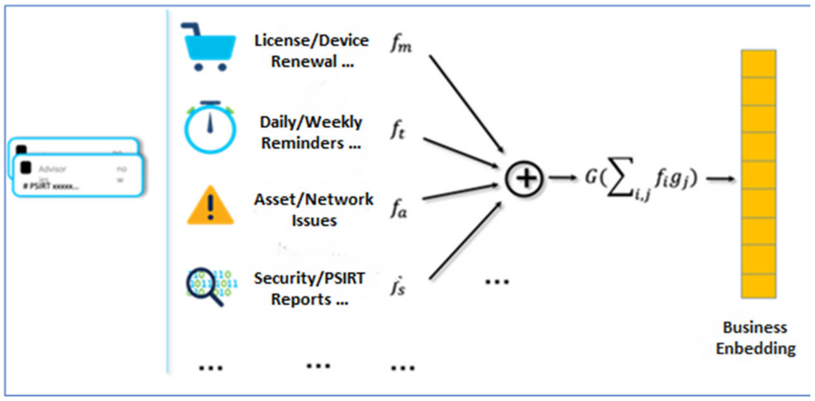

Figure 5a: Business Context Embedding Model

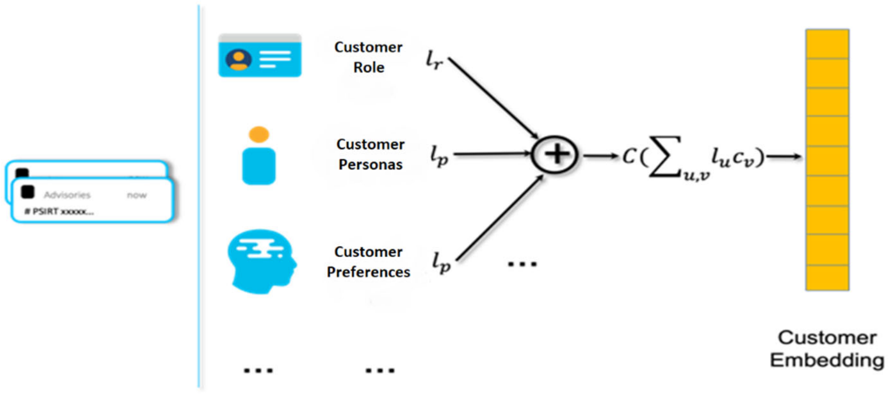

Figure 5b: Customer Behavior Embedding Model

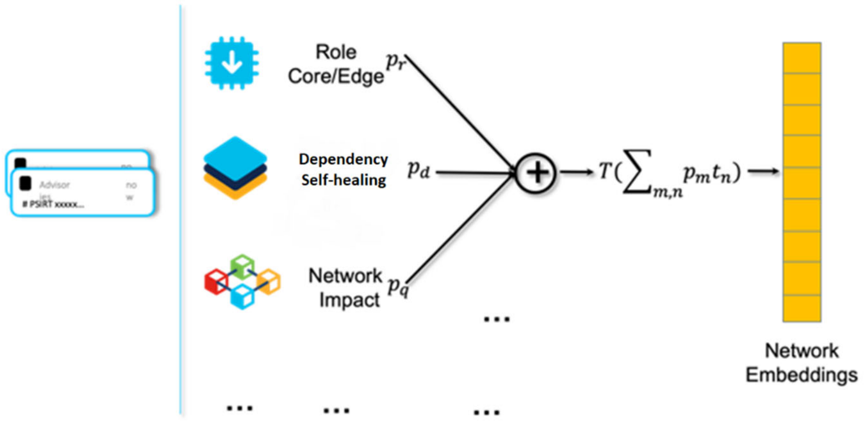

Figure 5c: Network Configuration Embedding Model

To embed the context features (that were described and illustrated above) in a vector space, domain-specific regression models may be employed.

In the regression models, heuristic rules may be used to pre-emphasize the notifications from multiple perspectives. For example, the longer that a critical notification or vulnerability has not been addressed, the more urgent a notification will be. Among other things, the monetary costs that are related to license and device failure may be also urgent.

Regarding business embeddings, such particulars may be obtained from the attributes of the notifications regarding a campaign, associated assets, and a receiver's business scope. Some additional information related to the service level and historical business record of the receivers may also be converted into business embeddings.

Regarding customer embeddings, customer features may be extracted from an individual's behavior. One example may encompass the click-through patterns and the response time patterns which may be used to reflect a customer's thoughts on different notifications. Other features (such as geographic features, personas, roles, and job duties) may also be utilized for these feature embeddings.

Regarding network context embeddings, features that are included in this embedding may be extracted from the operational and configured state of network devices. Metadata that is associated with a device (such as the business criticality and functional role of the device along with license details, configuration information, feature utilization, etc.) may play a critical role when the network operator chooses to perform a remediation action on one device versus another. Such network-specific embeddings may be used at the customer level as well (and interpreted globally) or at the vertical level to further establish patterns of remediation and action based on the role and function that a device plays in context to the business.

The generated domain embeddings (as described above) contain meaningful information regarding customer preferences, behaviors, interests, and network impact. Specifically, when a regression model is applied to embed multiple features, the dimension of the selected feature can reflect the diversity or density of a statistic's distribution. In general, the dimension of the features is proportional to their complexities.

Further, the domain embedding generation that was described above is based on the truth that for the first time the business, customer, and network configurations are incorporated to assess the priorities of the notifications. The highly customized evaluation system that is possible according to the techniques presented herein tailors the notification system to maximize a customer's experiences.

To model the mapping from input notification sequences to prioritized notification sequences, aspects of the techniques presented herein utilize a transformer that is based on a Seq2Seq architecture to learn a notification sorting. Here, all of the domain embeddings for each notification are considered as the semantic context to define the notification in a 'network' language. When training the model, the sequences of customer's click-throughs may be used as the GT of a notification's priorities. Unlike previous work that focuses on the regression of absolute priority scores, the techniques presented herein leverage the advantage of a Seq2Seq model to generate a relative priority sequence. For the incoming notifications, the Seq2Seq model may automatically learn the latent correlation and relative priorities among notifications in a certain time range.

Figure 6, below, depicts elements of the approach that was described above.

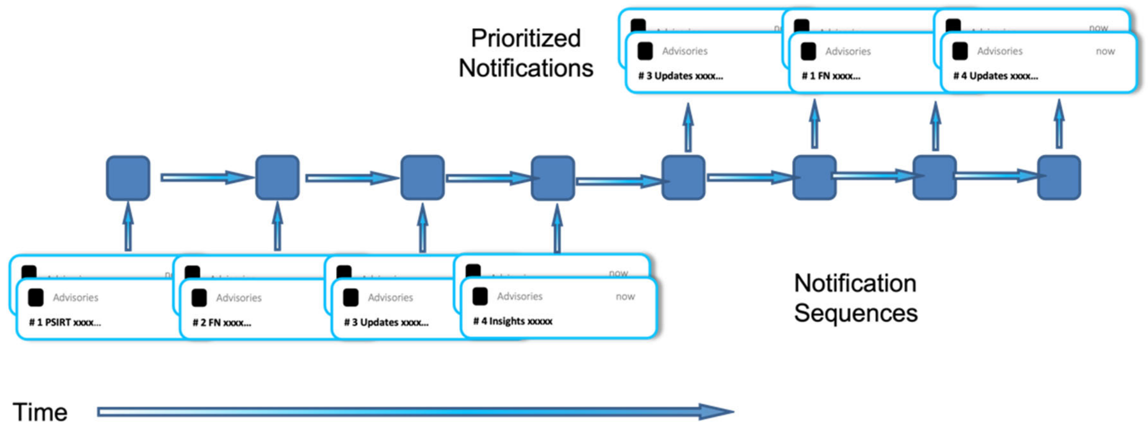

Figure 6: Transformer-Based Seq2Seq Model

As shown in Figure 6, above, a Seq2Seq-based DeepSorting model takes the context embeddings from incoming notification sequences and directly outputs the predicted click-through sequences. The sorting process skips the absolute priority score estimation of each notification but instead predicts the relative priorities. Compared with the absolute priority score estimation, the sequential priority prediction may represent the latent context and may be more robust in prioritizing emergent notifications. Specifically, a DeepSorting model jointly learns a customer’s click-through behavior in the temporal domain and, accordingly, may be more easily customized. Similar to language translation, such a DeepSorting model may inject sparsity into the output sequences, which further alleviates the risk of wrongly excluding important notifications.

According to further aspects of the techniques presented herein, to mitigate the discrepancy between the prioritized notifications and a customer’s final actions, action modeling may be used to generate actionable insights based on the predicted notification sequences. Figure 7, below, depicts elements of such an approach.

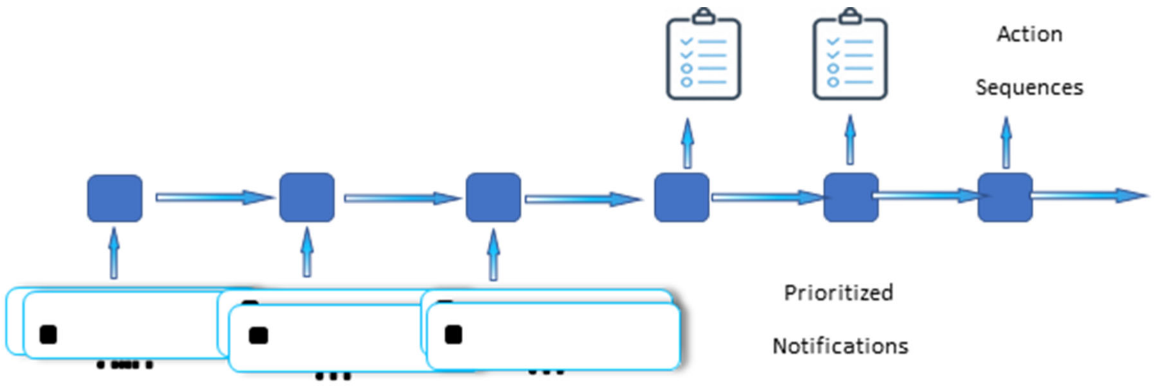

Figure 7: Seq2Seq Model Maps Prioritized Notifications to Customer Action Sequences

Since such a recommended action is learned from a customer’s historical data, the accuracy of the recommendations will be easily affected by the training dataset statistics. To ensure a robust and rational action recommendation, a continuous learning module may be added to the action modeling as described above in connection with Figure 3. Relying on lifecycle inputs, the action model may learn more comprehensive actionable insights.

As described and illustrated in the above narrative, use of the techniques presented herein offers a number of benefits. Those benefits include an innovative DeepSorting mechanism that supports a smart notification system that can prioritize the numerous domain notifications to optimize a customer’s actions more efficiently and economically; a prioritization of notifications based on business, customer, and network models top priority jobs; adaptive learning based on a customer's previous actions in response to various notifications; customer-centric modeling in the notification system to personalize the message distribution and recommended actions; an approach that is generalizable to a multiple factor notification sorting task; a reduced business risk and an improved customer experience through an optimization of the action priority; and, through the employment of a self-healing mechanism, DeepSorting deflects the task repertoire and saves operational cost.

In summary, techniques have been presented herein that support a process for intelligently filtering and prioritizing notifications to reduce noise and deliver recommendations that will provide the greatest impact to an environment. Aspects of the presented techniques encompass a smart recommendation notification system that prioritizes network actions based on multiple embedding spaces and dimensions; the use of business and financial logic, a persona, and a network operating state for reducing network actions into a prioritized output; the use of click-through and sequence mining to establish a ground truth of event prioritization of a network operator; an adaptive learning method for tracking proposed network recommendations to the final action that may be executed by a network operator; and a method for reducing multi-step recommendations based on an identification of the most efficient sequence of events based on a network operator implementation.