Supplemental Materials for CoreFlow: Extracting and Visualizing Branching Patterns from Event Sequences Using a Rank-Divide-Trim Approach

Authors Anonymized

Contents

Pseudo Code for the CoreFlow Algorithm

Algorithm 1 CoreFlow algorithm

```
input
S: a collection of sequences
n: a tree node in the branching pattern
m: minimum support
F: an ordered list of milestone events defined by user (optional)
procedure COREFLOW(S, n, m, F)
    if F is not empty then
        f ← first element in F
        add f as a child of n
        remove f from F
        S_0 ← sequences from S that do not contain f
        S_1 ← sequences from S that contain f
        for each sequence s in S_1 do
            idx ← index of first occurrence of f in s
            trim s from 0 to idx
        COREFLOW(S_0, n, m, F)
        COREFLOW(S_1, f, m, F)
    if size of S ¡ m then
        add exit as a child of n
        return
    else
        e ← top ranked event from S
        S_0 ← sequences from S that do not contain e
        S_1 ← sequences from S that contain e
        if size of S_1 ¿ m then
            add e as a child of n
            for each sequence s in S_1 do
                idx ← index of first occurrence of e in s
                trim s from 0 to idx
            COREFLOW(S_0, n, m, F)
            COREFLOW(S_1, e, m, F)
        else
            add exit as a child of n
            return
```

Comparing branching patterns generated using different ranking functions and datasets

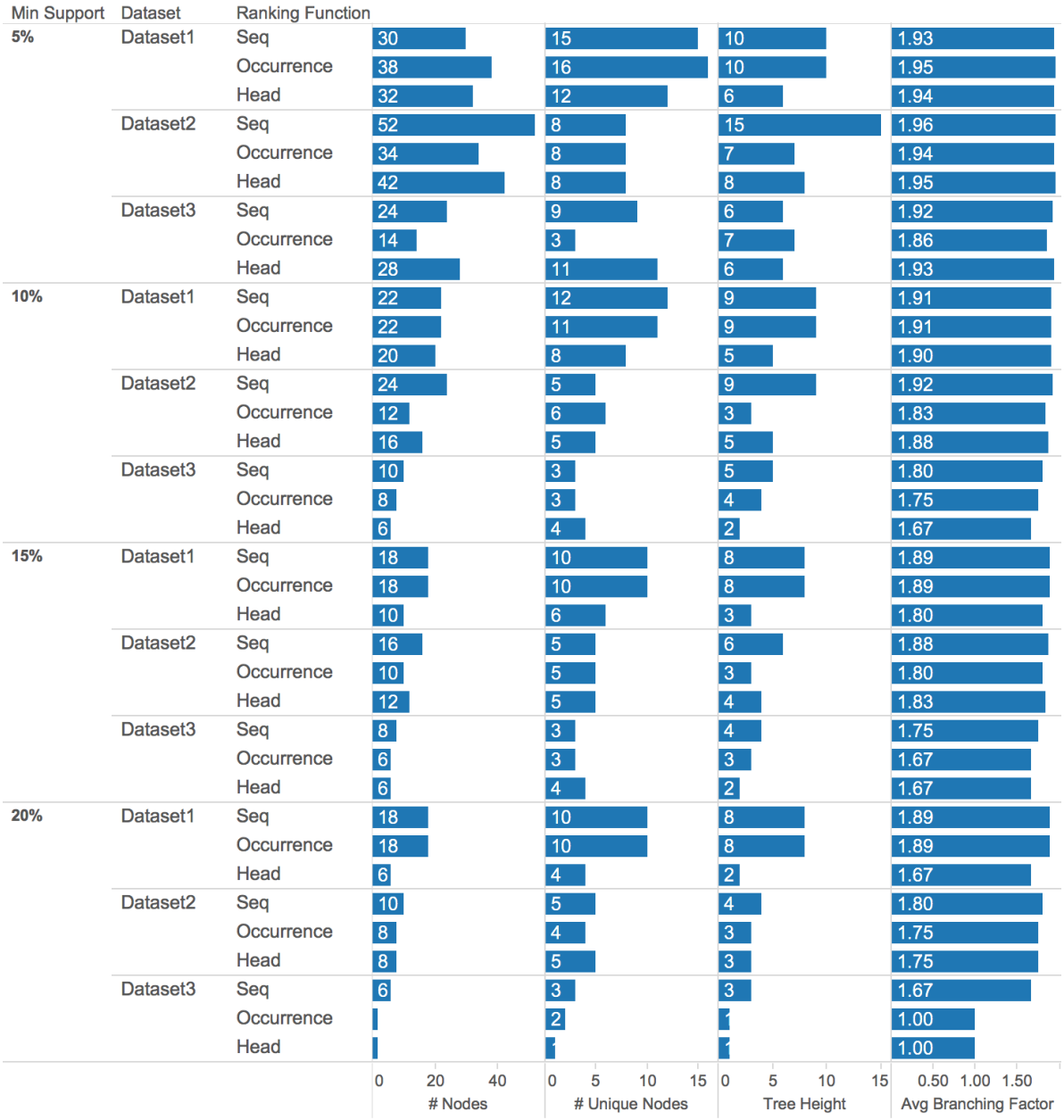


Figure 1: The properties of branching patterns generated using different ranking functions and datasets. We show four properties of branching patterns: number of nodes, number of unique nodes, tree height and average branching factor. The three datasets are the ones used in the evaluation study (Table 1). We use three ranking functions: 1) SEQ: the number of enclosing sequences as a metric to rank events. If a sequence has multiple occurrences of the same event, we count the sequence only once. 2) OCCURRENCE: the total number of occurrences as a metric, and 3) HEAD: the number of occurrences as the head of sequence. We also vary the minimum support from 5% to 20%. In general, HEAD tends to produce less nodes and shorter trees, while SEQ results in taller trees with more nodes.

Screenshots for Case Study 1

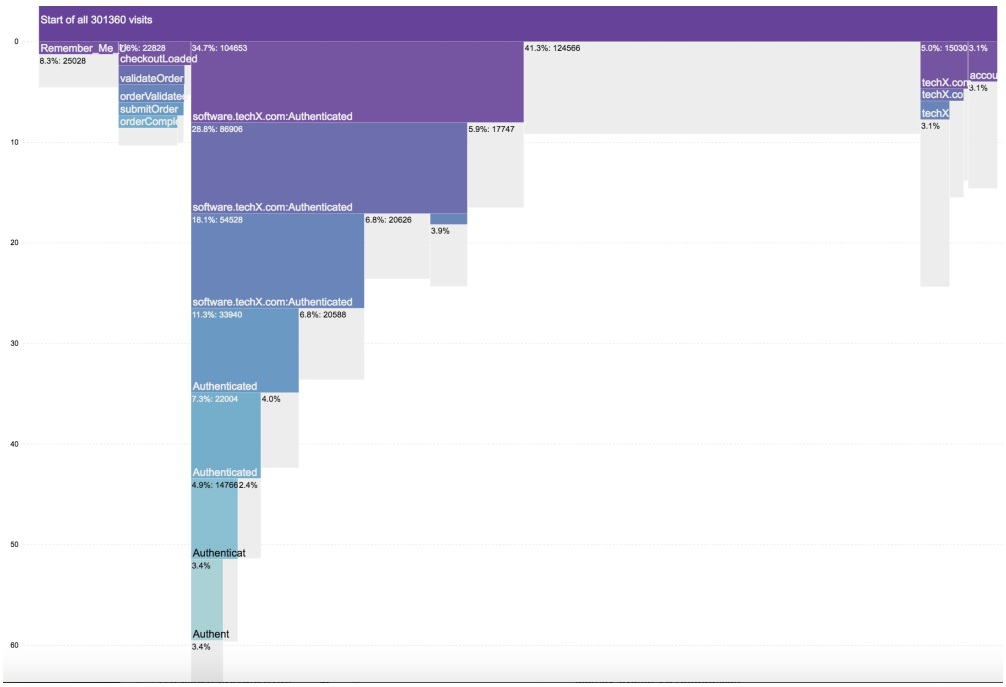


Figure 2: Initially, Brian thought that starting with as much data as possible was a good idea. He queried timestamped events on the company's website with a 6-month time frame, amounting to more than 300K sequences and 5 million events. The authentication page was identified as a repeating milestone event, and there was no frequent patterns for 41.3% of the sequences, indicating that the sequences were very heterogenous.

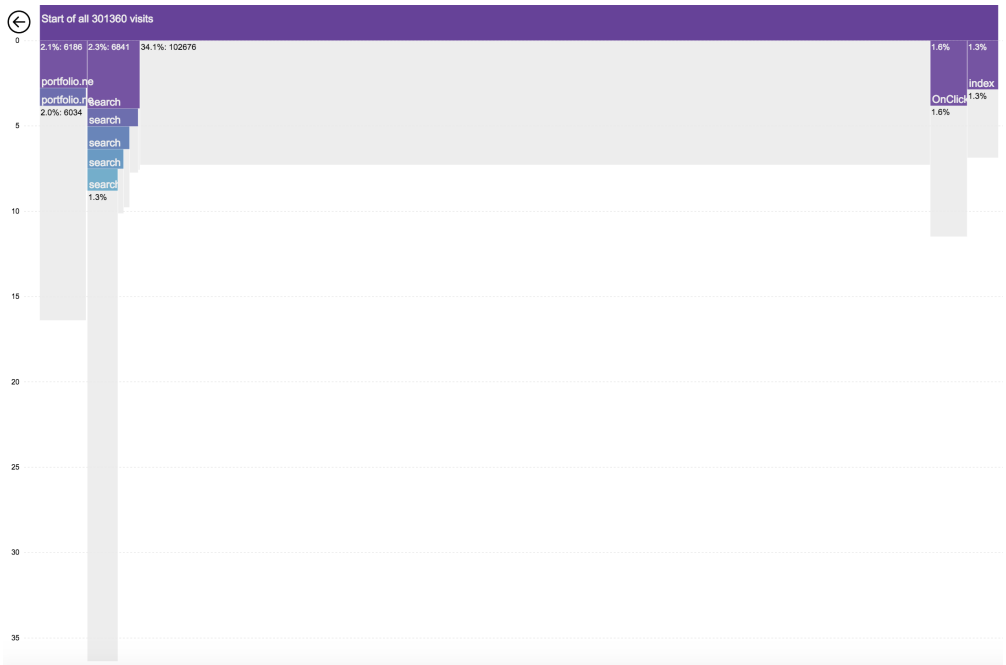


Figure 3: Drilling down to the 41.3% of the sequences uncovered search as a significant pattern, whereas 34% of the total sequences remain patternless. This prompted him to perform two actions: focusing on a particular product, and merging similar events into categories.

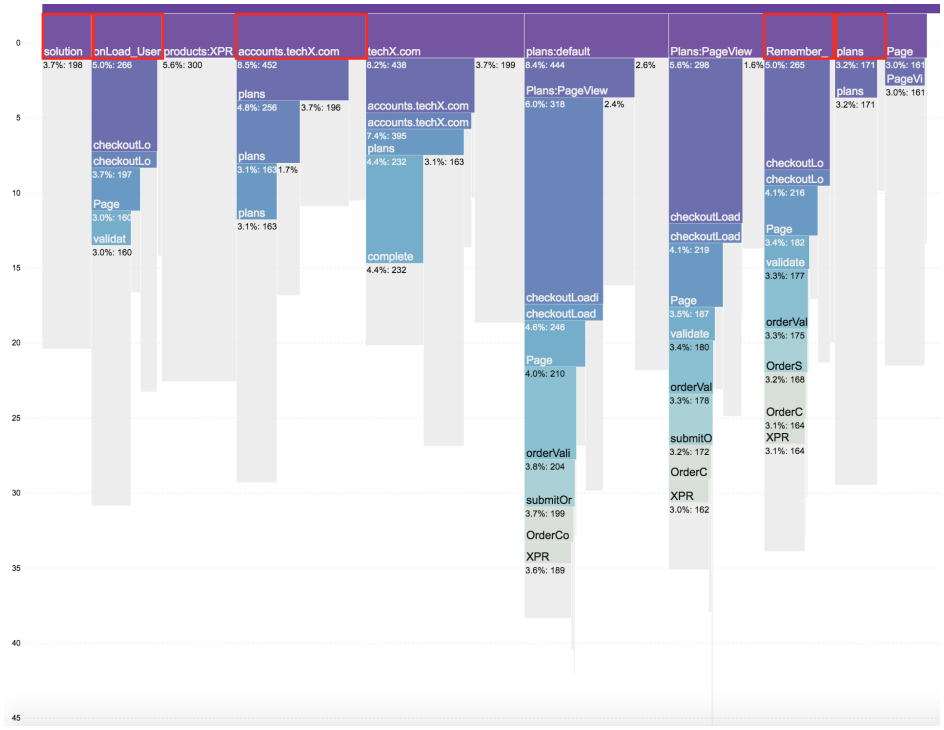


Figure 4: Brian grouped the sequences by the first events (entry events), and discovered that 25.4% of traffic were actually existing customers, based on the entry pages that were loaded first (these include “solution”, “onLoad_User”, “accounts.techX.com”, “Remember_” and “plans”, highlighted in red border). This insight helped him find a new user segment that he wasn’t tracking already. It turned out this ‘purchasers cohort’ was a large driver of direct traffic for the product in general. Understanding their behavior was critical to understanding the traffic on the website.

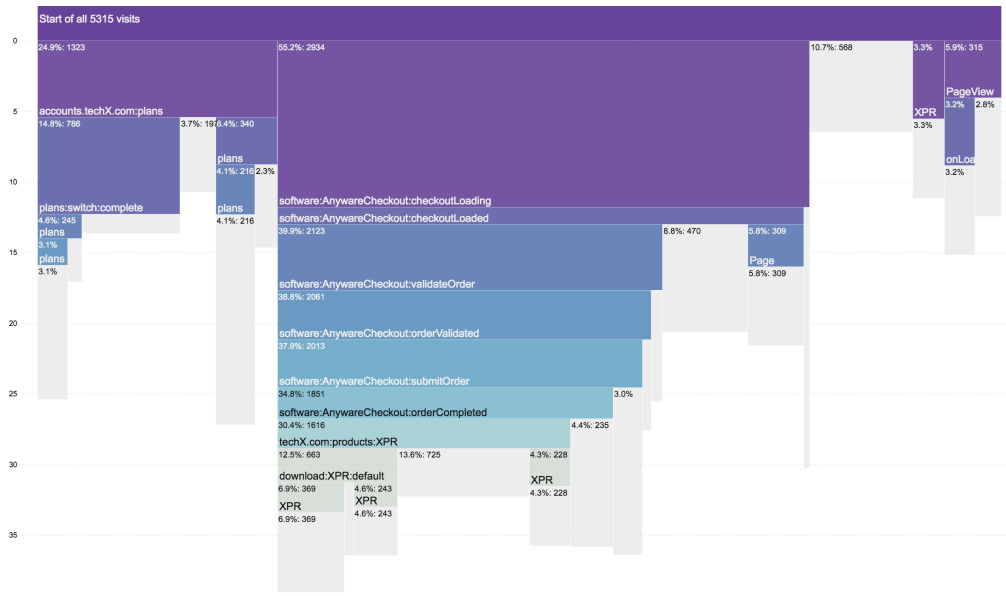


Figure 5: Brian was also able to see that of total traffic, 24.9% were switching from an existing plan (existing paid/free members) to the Single App option (highlighted in red border). This finding confirms what the company was seeing in the sales department. Many of the small and medium business customers transitioned to the main application they had been using in their full offering before.

Screenshots for Case Study 2

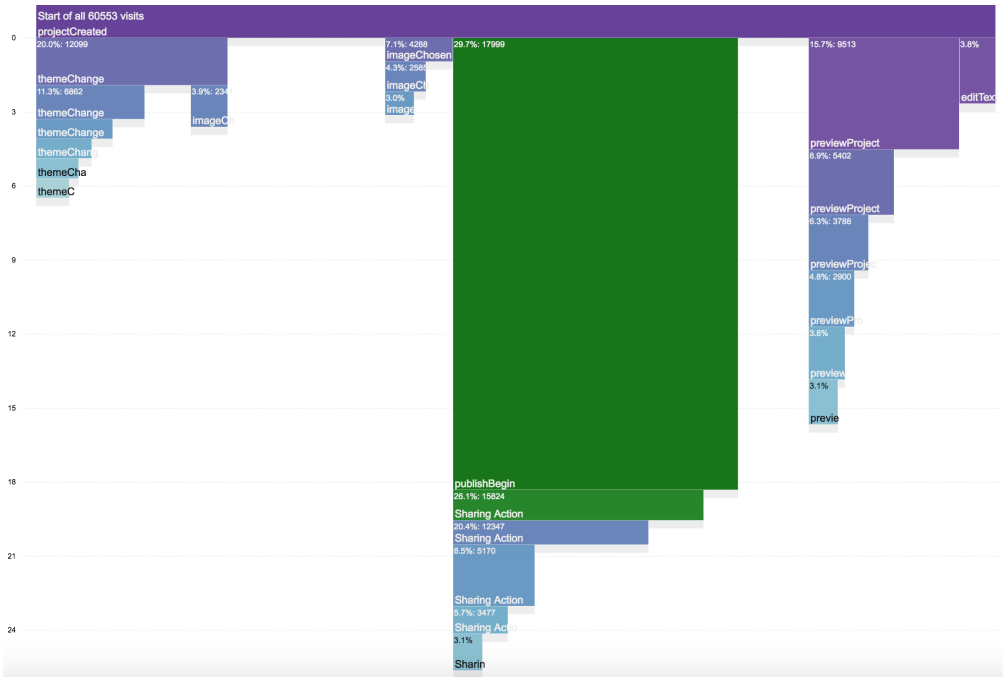


Figure 6: Using the funnel query, Stephanie saw that it took about 18 minutes for users to begin publishing and sharing their videos relative to the beginning of the sessions.

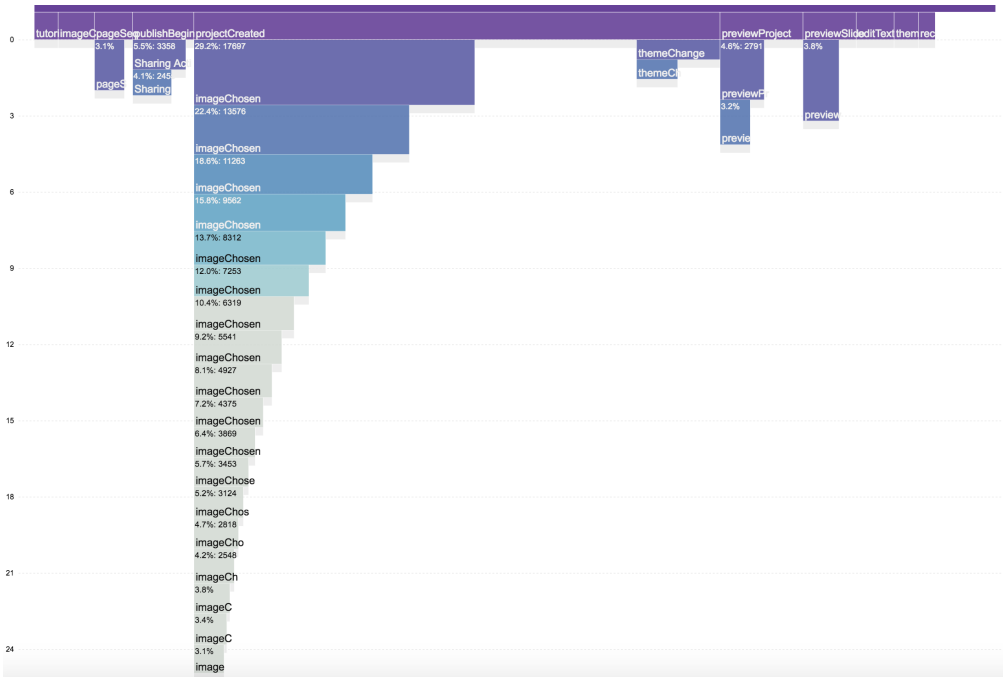
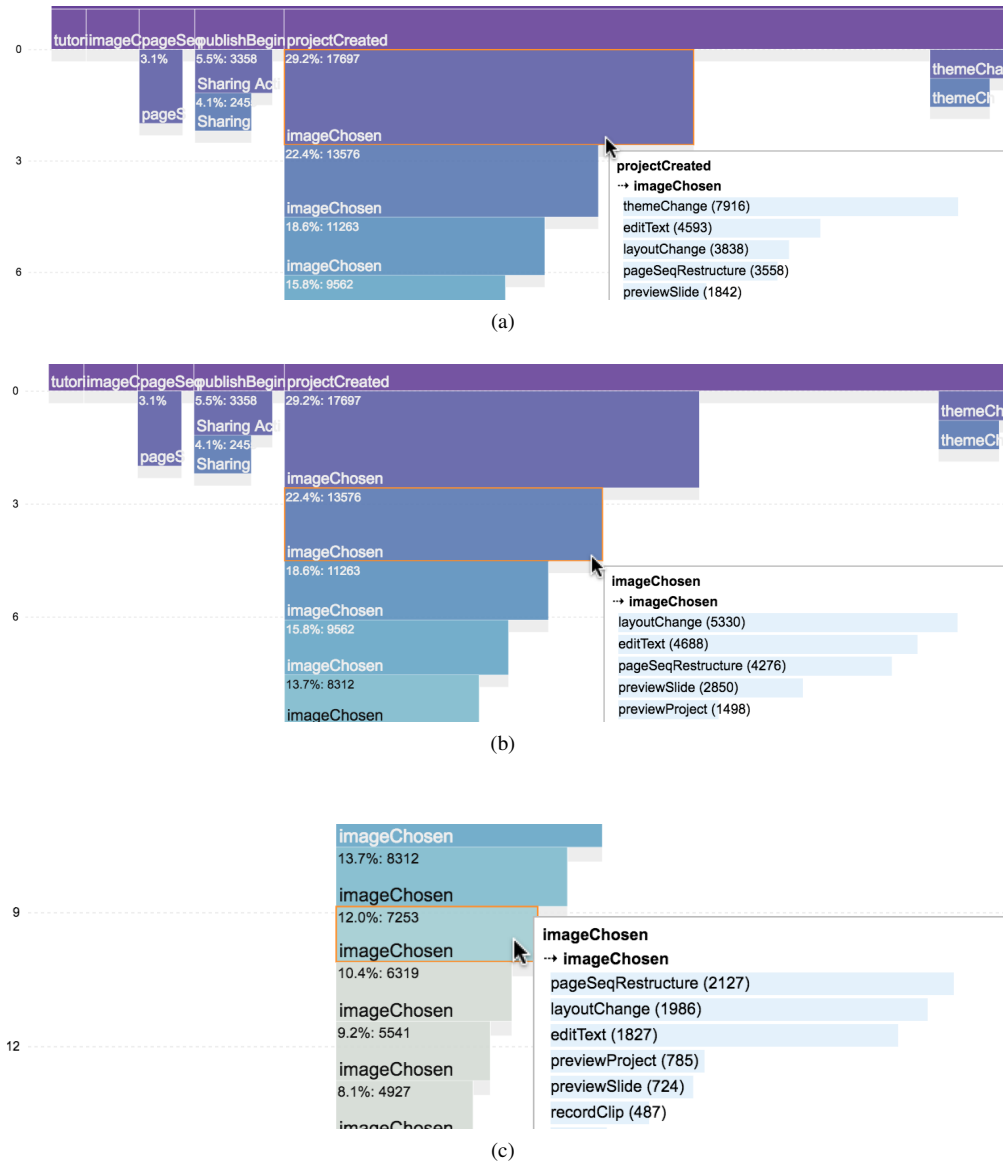


Figure 7: The most frequent user workflow was choosing images to embed in their videos.



(a)

(b)

(c)

Figure 8: Users were changing video themes, changing layout and editing text amid the first few image embedding operations. As they were making progress, they started to work on page sequence restructuring more, indicating a shift of focus.

Screenshots for Case Study 3

Figure 9: The initial overview showed that the marketing efforts on different media platforms eventually led 90% of the users to the homepage of the product, and on average it took about 30 days to achieve this result. After the event of landing on the product homepage, numerous searches took place on the site, which was not surprising since the primary function of the product involved searching for assets.

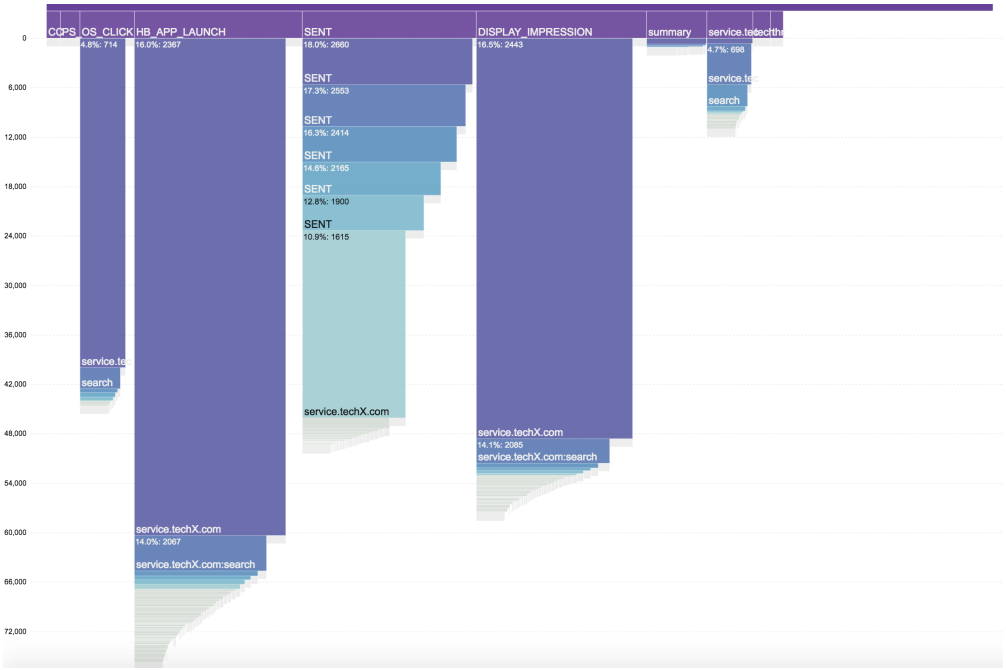


Figure 10: Jason asked if we can group the sequences by the first events. After we introduced this "group by entry event" feature in CoreFlowVis, he was excited because he could then compare customers' journeys by the first media touch point.

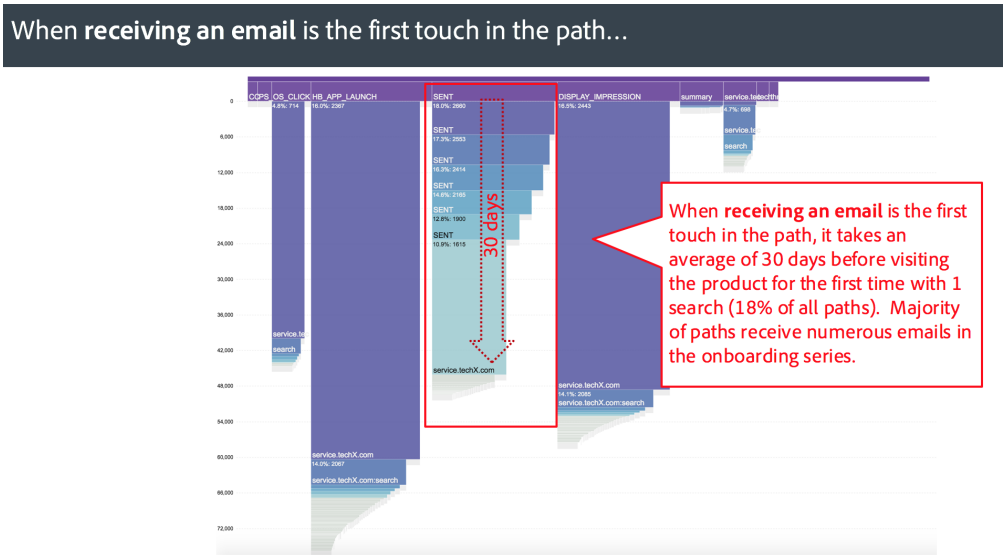


Figure 11: When receiving an email is the first touch in the path, it takes an average of 30 days before visiting the homepage for the first time with 1 search (18% of all paths). Majority of paths receive numerous emails in the onboarding series.

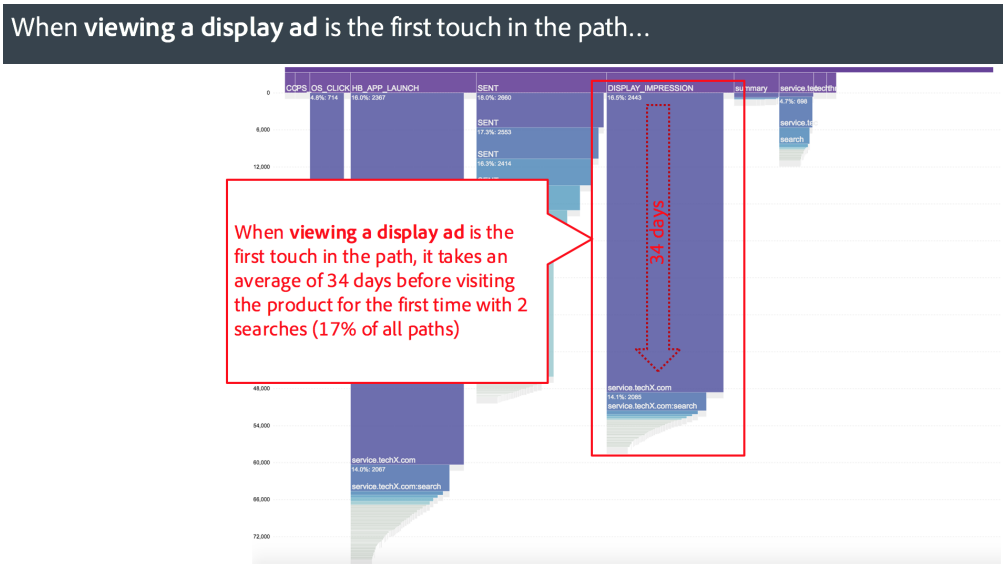


Figure 12: When viewing a display ad is the first touch in the path, it takes an average of 34 days before visiting the homepage for the first time with 2 searches (17% of all paths).

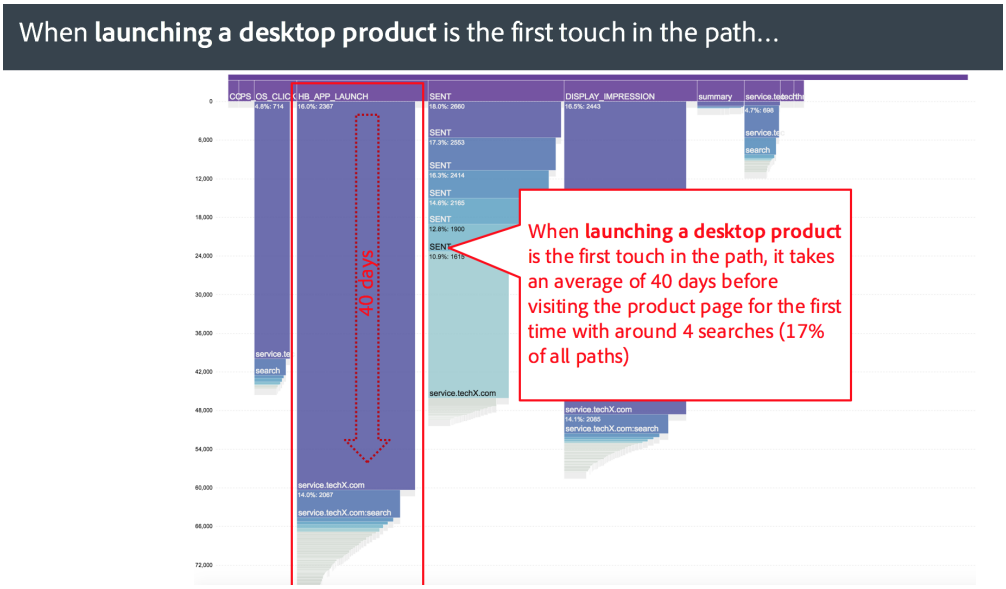


Figure 13: When launching a desktop product is the first touch in the path, it takes an average of 40 days before visiting the homepage for the first time with around 4 searches (17% of all paths).