On the Most Representative Summaries of Network User Activities

Joshua Stein[a], Han Hee Song[b], Mario Baldi[b,c,*], Jun Li[a]

[a] *University of Oregon, 120 Deschutes Hall, Eugene, Oregon, USA*
[b] *Cisco Systems, 255 W Tasman Dr., San Jose', California, USA*
[c] *Politecnico di Torino, C.so Duca degli Abruzzi 24, Turin, Italy*

Abstract

A summary of a user's Internet activities, such as web visitations, can provide information that closely reflects their interests and preferences. However, automating the summarization process is not trivial as the summary should strike a good balance between generality and specificity, while there is no gold standard for doing so.

In our approach to summarizing user information, dubbed SUM, we develop two scoring mechanisms that cooperatively optimize for polarizing criteria. After mapping user activity information onto a category tree, the scoring mechanisms highlight the most representative tree node (or summary); the node provides an aggregated view of the activities most characteristic of the user. We evaluate our approach by using web activity on the network of a large Cellular Service Provider and summarizing it to devise interests of individual users as well as groups. We compare SUM against an algorithm that discovers Hierarchical Heavy Hitter and show that SUM uncovers previously unknown information about users.

Keywords:
network user activities, user interest discovery, interest summarization

1. Introduction

As people are heavily connected to each other through the Internet more than ever before, they communicate a substantial amount of information about themselves. Such information is embedded in their network traffic, such as website visitations, data exchanged using different (mobile) applications, and GPS coordinates sent from their mobile devices. This data provides an opportunity to discover rich information about the users that can be of very high value to communication service providers, hence ultimately to users themselves. It is

[*] Corresponding author

Email addresses: `jgs@cs.uoregon.edu` (Joshua Stein), `hanhsong@cisco.com` (Han Hee Song), `mario.baldi@polito.it` (Mario Baldi), `lijun@cs.uoregon.edu` (Jun Li)

well known that the many services available free of charge on the Internet, such as search engines, social media, news, e-mail accounts, are offered in exchange for the user authorization to use the information she shares for commercial purposes (most commonly advertisement). However, one of the services that we are offered no other option other than paying for it, is access to the Internet. In fact, connectivity providers have no easy way of monetizing on data collected about their customers. While they are in principle in the ideal position as they can potentially access information shared through all of the specific services, such a vast amount of data is difficult to consume because it is composed of an extremely large number of very detailed items. A way to harness such a valuable resource would enable connectivity provider to give their customers the option of receiving fixed and mobile Internet connectivity free of charge in exchange for the consent for the provider to tap into the information that users exchange.

Summarization is the key to make such wealth of information manageable and practically usable, thus giving its users an opportunity to benefit from its value. The difficulty in summarizing the diverse sets of information is that there is no golden rule to objectively assess weights of different activities, i.e., how representative they are of users and their interests.

Example. *A content provider (CP) may be interested in understanding its users' cyber activities. When a user, Alice, shops for skates from an Internet shopping site, the CP may consider her interested in shopping, sports, or both. When new logs of Alice visiting a webpage of a ski resort come in, the CP may consider her to be into 'winter sports'. If new data reveals that she also frequents score boards of baseball, basketball, etc. on ESPN, the CP may consider Alice's interests to be simply in 'sports'.*

As depicted in the above example, summarizing a user's interests is far from being trivial. Some inputs may broaden the scope of a user's interests, some may narrow it down. Hence, a systematic method that strikes a good balance between generality and specificity is needed.

A natural approach to this problem is to depict it as a graph of interests and determine which interest is the most significant. The computation of significance falls under a class of problems known as graph centrality, with PageRank [1] being one method to solve it. However, methods such as PageRank are insufficient for computing the most significant node in a *structured graph* such as one where user interests are represented as a tree.

Research on finding Hierarchical Heavy Hitters (HHHs) addresses specifically the case of tree-structured data [2, 3, 4]. Methods for the identification of HHHs are particularly effective in summarizing activities in IP prefix-based *trie* structures. However, as their target applications are limited to network topologies where associations among tree nodes are strictly enforced by the IP addressing, these algorithms cannot be directly applied to our context where nodes are associated through softer, *semantic* similarities. Therefore, a more general approach is necessary for summarizing data categorized in ways less structured than IP prefixes.

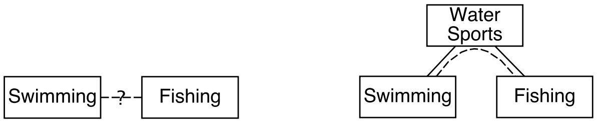
In this paper, we develop an approach, referred to as **SUM (Summarizer for User inforMation)**, that flexibly summarizes users' network activities into information about the users and their interests. To allow fair comparison among various network activities a user conducts (such as web browsing, usage of mobile apps, sharing of information), SUM maps them onto a single category hierarchy. Once the user activities are standardized, it then searches for the most representative summary of the activities within the hierarchy. For this, we develop two scoring methods based on Graph Centrality [5] — *choice score* and *stop score* — to perform a search on the category hierarchy. Beginning from the root of the tree, for each node, we assign a choice score which represents the preference of a direction, i.e., a child node, to traverse further. At the same time, we assign a stop score which is used in determining a sweet spot between choosing a general vs specific (i.e., deeper in the hierarchy) node on the branch the choice score chooses. Traversing the tree based on the two scoring mechanisms, SUM identifies a tree node that best represents the user's activity.

Challenges. The approach we developed for summarizing user information heavily depends on the structure of *data categorization*, which, being built by humans (*i.e.*, domain experts), is prone to be imperfect. For example, the tree might contain *inaccurate semantic structures* whereby children of nodes in the ontology tree may not be completely covered by the semantics of their parent. We do not place restrictions on the *topological properties of ontology trees*, *i.e.*, the number of children or ancestors a node may have. In addition, due to the subjectivity in determining what is a good summary, we *lack ground truth* to evaluate how well SUM summarizes user information.

Addressing the above challenges, we run SUM on a dataset of web visitations from a large CSP (Cellular Service Provider) in North America and map them onto an ontology with 65,000 user activity categories. We analyzed the dataset from the perspective of multiple different applications for our approach. To the best of our knowledge, this is the first work that systematically summarizes network user activities in a large scale. Our approach makes the following specific contributions:

- We design and implement the SUM algorithm that flexibly chooses a representative summary which has an appropriate balance between being general and specific. Hinged on the concepts of graph centrality, for a user (or a group of users), our algorithm determines the most representative yet specific summary from a pool of hierarchically classified activities. Our algorithm allows analysts to easily tune the summary to be between general and specific, depending on the application.

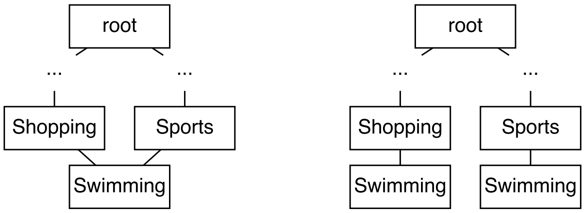
- We evaluate SUM using web browsing history collected from a large CSP covering 4 million web visits from 150,000 Internet users for five days. We demonstrate how SUM is able to summarize the web browsing interests of individuals as well as groups of users. The summaries that are produced by SUM have high stability under varying amounts of interest data differing by 1.37 categories on average, and possess a common depth (i.e., specificity) of 2.42 on average. Furthermore, the summaries produced by SUM are relevant even in the presence of skewed interests.



(a) Flat Category

(b) Tree

Figure 1: Expressiveness of categorical semantics. Dashed line indicates an implicit relationship.



(a) Graph

(b) Tree

Figure 2: Explicit information of categorical semantics.

- We perform a case study with SUM and HHH on a collection of web browsing activity generated by a large number of Internet users. SUM and HHH agreed on many of their most specific categories, but SUM proved to have more descriptive and representative summaries than HHH. SUM revealed properties of the users' Internet usage, particularly honing in on their preferred search applications. Beyond search, users were found to have a particular affinity for visiting footwear shopping sites.

2. Background

In this section we define the components necessary for summarization in addition to the properties a summary should exhibit. The summarization process requires data, to which we add annotations in the form of the category of an ontology the data may fall under. In this work the semantics of the annotation is restricted to be defined by an ontology composed of categories the data may fall under. The summary that results from the data must be general enough to capture a good fraction of the data, while still being specific enough such that no significant information is lost. In other words, building a summary takes up the challenge of striking a balance between the breadth of data captured, as well as its depth.

2.1. Categorization Ontology

The ontology the summarization process operates on has relationships between categories. The number of relationships for a category is not restricted, but the topology of the ontology must be a *tree*. The motivation for using a tree

URL	Category
telegraph.co.uk/news/... .../worldnews/middleeast	Regional >Middle_East >News_&_Media
www.youtube.com	AudioVideo >Video_Streaming >Community_Video
www.radioreference.com	Business >Telecommunications >Two-Way_Radio

Table 1: Example input data with their category labels. Categories go from general to specific from left to right.

over flat categorizations or a general graph is due to the explicit relationship between categories and expressiveness of the topology, respectively.

Tree-based vs. flat categorization: Flat categorization, or keyword categorization, explicitly states the category that data falls under. The limitation of utilizing a single keyword is that keywords do not relate to one another, which would enable us to derive strong connections between data items. If a strong connection were present between single keywords then we would be able to construct a structured categorization, which we will cover more generally below. For instance, categories such as “swimming” and “fishing” may be used for labeling data but they lack any indication of relationships between them (*i.e.*, Fig. 1a). Conversely, in a tree topology having a single ancestor, such as “water sports”, expresses implicitly (and compactly) a relationship between the two nodes (*i.e.*, Fig. 1b).

Tree-based vs. general graph-based categorization: A general graph is able to accurately fulfill the expressiveness of an ontology tree as well as more complex relationships. A category could be reachable through multiple paths from the most general category (i.e., the one at the root of the tree) and thus obtain different meanings depending on the path. An example of this would be reaching the category “swimming” from a “shopping” category versus a “sports” category (see Fig. 2a). Although the path taken is informative, having alternative paths introduces several challenges. For each item of categorized data it is necessary to know not only the category it belongs to, but also the path that was taken during the categorization process, otherwise the correct semantic associated to the category (and the data) is lost. To somehow capture the information embedded in multiple paths to a category node from a root we allow redundancy in our ontology tree (*i.e.*, a node “swimming” can be present in both branches in Fig. 2b). Redundant categories placed in appropriate locations within a tree force the meaning of a categorized data item (in terms of path followed during its categorization) to be explicit.

2.2. Data Categorization

A crucial preparation for applying the proposed summarization technique is that the input data is categorized based on an ontology that fulfills the description in Section 2.1. Although data categorization is not within the scope of this paper, we assume that for any given data item the categorization process will output an accurate category. An example of accurately categorized data is found in Table 1 since each category is relevant to the given input URL. The categories for each example URL are very specific, yet the categorizations do not provide inaccurate categories. Because the degrees of specificity in the categorization is tightly related to the quality of summarization, using well-categorized data is key to the outcome of the proposed algorithm.

2.3. Properties of Summarization

The goal of our summarization is to discover the most specific category, yet representative of a large number of data items. Specificity and representativeness, which we will expound on further, are potentially contradictory in that a category that is specific may be representative only of a small subset of the data items, whereas a category that is representative of a large fraction of data items may be too general, i.e., does not provide any relevant information about the represented data items. Because the properties of being specific and representative are complementary, in order not to lose coherence we propose an algorithm that alternates two scoring mechanisms within.

The properties of being specific and representative are complementary and we lose coherence if they are discussed separately. The category chosen for summarization should be as specific as possible, while representing the dataset without losing a substantial degree of information. This *core* category is the essence of the dataset and as such a certain fraction of the dataset may be extraneous to it. In our algorithm design, we use the concept of graph centrality to discover the core category. And in our evaluation, we quantify the representative power of the core category based on the categories around it.

3. *SUM* Algorithm Design

Our algorithm is composed of four phases. Each phase is responsible for handling a challenge of summarization in isolation such that the following phases can assume the data has certain properties. We explain the mapping of web service visitations into the category tree in Section 3.1, the assignment of initial scores, dubbed original scores, to individual tree nodes in Section 3.2, the propagation of the initial tree node scores throughout the tree into new scores in Section 3.3, the summarization process in Section 3.4, and then the complexity analysis in Section 3.5.

3.1. Mapping Categorized Data to the Tree

In this first step, we take categorized network activity data of users, and insert the data into the category tree. Each node of the tree where data is inserted is labeled with a category (*e.g.*, Fig. 2b). The data, by previous assumptions, is mapped into the most specific category node of the tree. The result of insertion is that the data is now aggregated into their corresponding categories in the tree. In our application, we translate website visitation logs of users into an ontology tree with categories of web pages. The results of this step are URLs of websites that user visited mapped onto a category tree.

The data source may act as an extra dimension during the insertion process. Depending on the particular application of our summarization, we may either summarize activities of individuals or aggregates of a group of users. In the former, we map activity data of n users onto n separate copies of ontology trees. In the latter, we map activity data of n users (considering the group size to be n) onto a single ontology tree.

3.2. Original Score Function S

The original score function S is a function that translates the data that is inserted into the tree into a non-negative numerical score. The score is computed locally for each category (*i.e.*, tree node) and this local score is directly related to the magnitude of the importance of the data mapped to the given category. The original score function defines the properties we seek to summarize on. In our application, the original score function associates frequency of the URL accesses onto the nodes websites are mapped to.

The original score function is also able to be used for weighting or normalization. Weighting is a purely local form of normalization where data sources are given different levels of significance—significance based on knowledge of the collection process. Weighting data sources is therefore dependent on the data sources and that data themselves. Normalization, which is global, may be performed such that scores across the tree have a certain property. This process is non-trivial due to complications associated with the tree itself, the data sources, and the categorization process. It is therefore recommended that the original score function be kept simple for the purpose of comprehension. We consider the following instances of the original score functions that have clear utilities:

- **Summation of activity**: This original score function counts the frequency of activities associated with a category on the tree and is able to discover categories that users are biased towards. Summing the activity within a category, such as when we count the number of visits of a website, allows for us to learn about the raw website visitation patterns of the users in our study. It is unlikely that a single user would produce enough activity to skew the results, which may optionally be prevented through normalizing the activity of each user, and so the summation also serves as a metric of visitation popularity.

- **Number of users with the activity**: This original score function counts the number of users with activity associated with a category in the tree. This approach removes all bias caused by users with biased activity and instead looks at popularity across the user base. We utilize this approach so that we capture the activities that are performed by users no matter their visitation history. Activities that have a wide range of users yet low visitation history, and potentially fewer re-visitations than the previous approach, allows us to learn about potentially more universally popular activities.

- **Logarithm of the summation of activity**: This original score function applies the logarithm function (ln in our case) to the count of the frequency

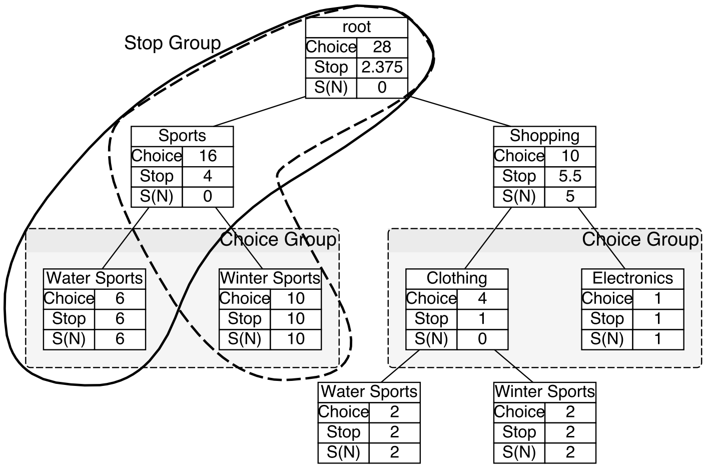


Figure 3: Calculation of choice and stop scores given a single original score function. The scope of usage of the stop score is shown as being between a parent and child, whereas the scope of usage of the choice score is shown as being between siblings. $\beta = 0.25$, G = sum of activities.

of activities associated with a category in the tree. The logarithm of the summation of activity is another method for removing bias caused by some of the users. The primary difference compared to the previous approaches is that this approach allows for discrepancies between the amount of activity between users. In effect this approach maintains the properties of summing the activity within a category, except for the fact that the logarithm grows slowly. The information that is then extracted from this approach is able to capture the interests of users with more general interests.

3.3. Two score propagation

We then propagate original node scores computed in the previous phase up the tree. This phase is necessary to fill gaps within the tree and provide information at ancestor nodes for decision making. Our method for score propagation involves the propagation of two scores, one for *branch selection* and one for the *level of specificity* dubbed the **choice** and **stop scores**, respectively. The purpose of the **choice score** is for comparing related subcategories against each other (*i.e.*, to **choose** a child among its siblings). The purpose of the **stop score** is for comparing a category's score to those of its subcategories (*i.e.*, to **stop** propagating into a branch if the category's score is higher than that of its subcategories). The entire score propagation process is depicted in Figure 3 in addition to the scope of the stop score and choice score which is denoted by their respective groups.

Term	Definition
$F(i)$	Propgation function (defined as Equation 1)
$G(s)$	Accumulator function over the set s
$S(i)$	Original score function of node i
$C(i)$	set of children of node i
A	Adjacency matrix
β	Damping factor

Table 2: Definition of terms.

3.3.1. Advantages over a single score

Propagating two scores separately helps quantify two incomparable properties (*i.e.*, significance of a category among its siblings and significance of a category compared to its subcategories) that are otherwise difficult to be aggregated into a single score. Alternatively, if we used a single score to encode both properties, it would become infeasible to compare the respective properties as we would have lost information in the process. For instance, if we consider the sample scoring in Figure 3, we can see that if we utilized the stop score for our choice of a category at the top level, we would choose "shopping" (that has the highest stop score at 5.5 vs 4 for "sports"), while the choice score indicates "sports" (that has the highest choice score at 16 vs 10 for "shopping") as being the most representative category. Also the choice score cannot be used by itself. In fact, since it provides an indication of significance among siblings, its value cannot be used for comparisons across the whole tree because it propagates independent of depth. For example, in the sample scenario depicted in Figure 3, the root has the highest choice score at 28 and it would always be chosen over any other node. A single score fails to represent independently the component of the score that is contributed by the choice score or the stop score. Even after normalization shown in Section 3.2, it would become necessary to split the scores into their respective components for the purpose of comparison thereby invalidating any advantage of a single score.

3.3.2. Propagation formula F

Score propagation in our algorithm is a central concept. Here, the original scores are propagated from the leaves all the way to the root as recursive functions. While any recursive function could feasibly work for propagating scores throughout the tree, we root our algorithm from the concept of graph centrality as it has been proven to be effective in discovering the most important (or 'central') vertex in a graph [6] Among many implementations of the centrality measures, we use Katz centrality metric [5]. Different from simpler measures that only considers either a single path (*e.g.*, shortest path [7]) or immediately neighboring nodes (*e.g.*, common neighbor or Adamic-Adar measures [7]), this random walk-based algorithm builds a more comprehensive perspective of node centrality by considering all paths to all the other nodes in the tree. Formally, Katz centrality of a node i is

$$Katz(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \beta^k (A^k)_{ji},$$

where n is the number of nodes in a graph, j are the nodes being compared to i. The formula iterates through all paths with length k where $k = 1, \cdots, \infty$ using an adjacency matrix A (where $a_{ij} \in A = 1$ if a vertex exists between i and j, 0 otherwise). As it is shown in the formula, the Katz value is additive to the number of paths while the value of each path is multiplicatively penalized by a damping factor β with respect to the path length k.

Based on the above theory, we now define our function for score propagation as follows. Let i be a category node and $C(i)$ be the set of children of i. Furthermore, let $S(i)$ be the original score function for i, defined in Section 3.2. The score propagation function $F(i)$ is therefore

$$\begin{cases} S(i) & \text{if } C(i) = \emptyset \\ S(i) + \beta \cdot G(\{F(p) | p \in C(i)\}) & \text{otherwise} \end{cases} \tag{1}$$

where $G()$ is a function that accumulates multiple scores into a single one. A simple implementation for $G(\{F(p)|p \in C(i)\})$ is a summation over all children of a node of the score propagation function computed for each one of them. β is a tuning parameter for weighting a category's own score and the aggregate child score, respectively. In order to avoid score explosion, $\beta \leq 1/ \parallel A \parallel_2$ where $\parallel \cdot \parallel_2$ represents $L_2 - norm$. An in-depth discussion on β will follow in Section 4.

Our algorithm requires score propagation to occur for two different scores: the *choice score* and the *stop score*. Each score, as we describe below, contributes certain properties to the summary.

3.3.3. Choice Score

The choice score is one of the two scores that is computed during score propagation. We assign the choice score to each sibling under a given node; the subtree rooted in the sibling with the highest choice score is selected as containing the summary.

The motivation for the choice score is that categories are not guaranteed to have either equal depth or to progress in specificity at equal rates. The choice score is then computed such that we can determine which of the subcategories, *i.e.*which subtrees, is the most significant.

We propagate the choice score using a function in the form of Equation 1. Recall that each subcategory has variable unrestricted depth and that data is only mapped to categories that are as specific as possible. We avoid bias from data at varying depths by choosing $\beta = 1$ so that the choice score is calculated irrespective of depth. Optionally, bias may be introduced for data mapped either higher or lower in the subtree by selecting a $\beta \neq 1$.

3.3.4. Stop Score

The stop score is the second of the two scores that is computed during score propagation. The stop score, unlike the choice score, is not used to compare sibling subcategories against one another since the stop score is designed strictly for comparisons based on depth. As shown in Figure 3, the stop score along branch “root - sports - water sports” only compares the nodes on the branch but not “ball sports” or nodes in “shopping” branch.

The stop score uses the same function template as the choice score for propagation. The primary difference is that unbounded score growth is undesirable as that would result in a general category’s stop score consistently being greater than its subcategories. Ideally, we want the stop scores along the path defined by the choice scores to have an inflection point after propagation, i.e. stop scores increase up until reaching the correct category after which point scores decrease. We therefore use a $\beta \ll 1$ to ensure that stop scores do not grow unbounded.

3.4. Searching the Representative Node—Summarization

Algorithm 1 RepresentativeNodeSearch(node)

```
summary = node.category
maxStop = maximum subcategory stop score
maxChoices = set of subcategories with the maximum choice score
if node.stopScore < maxStop then
    summary = $\cup_{c \in maxChoices}$ RepresentativeNodeSearch(c)
end if
return summary
```

After we have propagated the choice score and the stop score throughout the tree, we are now able to perform our search of the most specific yet representative category, *i.e.*, our summarization. The algorithm provided in listing 1 starts at the root and recursively searches until it finds the correct category. During the search process, the algorithm utilizes the stop score and the choice score at every stage in the algorithm. Namely, the stop score determines how deep into the tree we look into, and the choice score determines which node to take at every branching points.

We now break the algorithm into its two primary components: the stop condition and the choice decision.

Stop Condition: After score propagation every category on the tree has an associated stop score. The algorithm compares the stop score of the current category to the maximum stop score of its children. If the current stop score is greater than the maximum child stop score, the algorithm then decides it has reached the correct level of specificity, and it returns the current category. Otherwise, at least one child has a greater degree of specificity, and the algorithm will continue its recursion, for which the algorithm makes a recurse decision as described right below.

Choice Decision: The choice score, as was described in an earlier section, is designed to compare subcategories of similar specificity. The algorithm can decide which subcategory to continue its search by determining which subcategory has the greatest choice score. If multiple subcategory carry the same choice score, then it performs recursion on all of them simultaneously.

3.5. Complexity Analysis

The algorithm we have described is split into score calculations and a summarization (search) function, each of which having different execution complexity characteristics. These two operations deal with different aspects of the ontology tree structure. For the purpose of describing the run times of the two operations, we define d and b as the depth and branching factor of the ontology tree, respectively.

Insertion: Inserting new values into the tree is required in order to ensure it reflects the actual user activity. Insertion complexity is $O(d)$, but it can be reduced (trading for additional storage requirements) to $O(1)$ by hashing into the tree.

Score Calculation: Score calculation occurs after new values are inserted into the tree. However, it does not necessarily need to be run after every single insertion. Depending on the specific application it might be run periodically or on-demand when a summary is needed. If scores are computed after a single insertion, the computation complexity is $O(bd)$: given that the score propagation function, F, includes the children as part of the computation of the score of a node for $\beta > 0$, the insertion of a value requires recomputing the score of all nodes on the path to the root, accessing all children of each one of them. If the score calculation is triggered after multiple insertions (e.g., periodically or on-demand), it must be applied to the entire tree, i.e., the complexity is the size of the ontology $O(b^d)$. Since less frequent calculations have higher complexity, the most appropriate timing shall be determined depending on the requirements of the specific application of the algorithm, in terms of how often the summarization is needed.

Summarization: Summarization occurs after score calculations. The operation starts at the root and must look at all children at each step, to select one and continue with the subtree it roots. This results in an execution complexity of $O(bd)$. It is possible to improve the run time by storing in each node the largest stop score among the children and a pointer to the child node with the largest choice score during score calculation. This can improve the run time to $O(d)$ because determining at each node the subtree that contains the most representative summary does not require to explore all children, but it introduces a $O(b^d)$ memory overhead for the additional information stored in each node.

4. Evaluation

One of the challenges in the evaluation of SUM is the lack of a ground truth. Consequently, the performance of the algorithm cannot be measured in the typical terms of precision and recall. Instead, we resort to alternative ways of assessment. First, we test the algorithm in bespoke scenarios that are

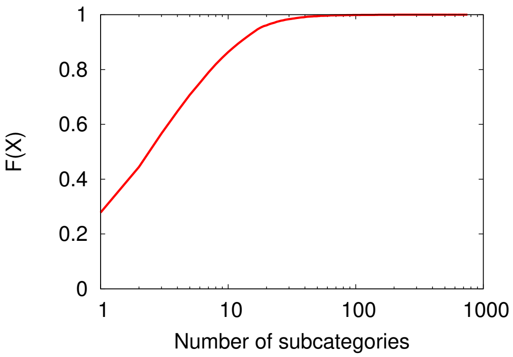


Figure 4: CDF of the number of subcategories per category in the tree.

simple enough to be manually analyzed to verify that the behavior matches what we would expect (Section 4.2). Then, we study the properties of SUM when operating on a real-life dataset (Section 4.3). Finally, we compare results produced on such dataset by SUM with the ones produced by Hierarchical Heavy Hitter (HHH) (Section 4.4). We conclude the section with a discussion of the limitations found in SUM (Section 4.5). Before delving into the assessment, we discuss the evaluation framework, including our real-life dataset and choices for some of SUM parameters (Section 4.1).

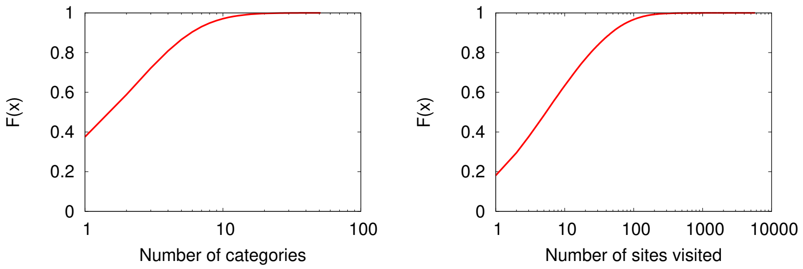
4.1. Evaluation Framework

4.1.1. Dataset

Category Tree. The ontology we used for categorization is the same as the one utilized by Alexa [8] with only minor variations [9]. Alexa uses a hierarchy of categories for the purpose of classifying websites into a category as specific as possible. The Alexa category organization allows for categories in different subtrees to reference each other. The modified tree we use ignores such references, i.e., each subcategory has a single parent category; moreover, we added a new categorization for online communities, *i.e.*, social networks. The hierarchical ontology has a total of $65,634$ categories, $53,654$ of which are leaf categories, i.e., they do not have any more specific sub categories.

The tree is highly imbalanced with a maximum depth of 6 and some leaves at depth as low as 4. As shown in Figure 4, the number of subcategories of a single category varies largely and is unrestricted across the tree. The extremely large number of subcategories may be attributed to degenerate cases, such as listing every country in a certain part of the world as subcategories. The properties of the Alexa-based tree used in our experiments are consistent with those outlined in the previously presented challenges.

Web Activity Trace. In order to evaluate our approach we used a web activity trace produced by analyzing traffic on the network of a large Internet service provider in North America for 5 days from a Monday to a Friday. Each instance of web activity, i.e., each visited website, is mapped on a category as specific



(a) Distinct categories visited by each user(b) Total number of website visits per user

Figure 5: Cumulative distributions of user web activity

as possible, i.e., as far as possible from the root of the Alexa-based category tree. The mapping process is not a contribution of this work; we use an existing approach [9] and deploy the resulting categorization of each visited website as the starting point for the application of SUM.

The $4,390,365$ web visits in the trace are generated by $150,688$ distinct users. We excluded on-line social networking activity from the trace used for the evaluation because virtually all users visit websites falling in this type, thus not being relevant with respect to the goal of SUM, i.e., summarizing browsing activity across users. The remaining activity includes $2,545,911$ web visits by $134,462$ users.

Since we are interested in users that are highly active across diverse categories, we studied web visitations. Figures 5a and 5b show the CDFs for the number of distinct categories accessed by each user and total web visits of individual users, respectively. Visitations to two websites mapping to a category and one of its subcategories, respectively, are considered as activities in different categories. We selected for our evaluation the subset of 922 users that have visited at least 100 websites across at least 15 distinct categories.

4.1.2. Scoring Functions and Parameters

In the experiments to evaluate our summarization approach we utilized similar functions for the propagation of both choice and stop scores with differences in the underlying original score function and parameters. The function G in equation 1 is set to the summation of the child scores for the computation of both the choice and stop score. In the computation of the choice score we set $\beta = 1$, whereas for the stop score $\beta = 0.25$. The selection of $\beta = 0.25$ for the stop score is motivated by our observations which will be discussed in Section 4.2 and experimentally confirmed in Section 4.4.1. Since the original score function returns the number of data items that are mapped to the given category, the choice score leads to choosing the subtree with the most data.

4.2. Evaluation in Bespoke Scenarios

In order to work around the lack of a ground truth, we build constrained scenarios that are limited to trees of a manageable size, compute the outcome of

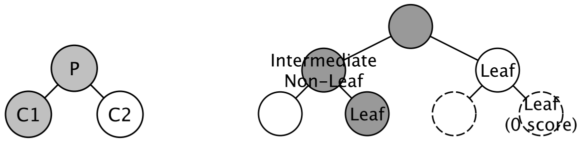
Original scores	C1=C2=0	C1>C2=0	C1=C2 >0	C1>C2>0
Parent selection	-	β >1	β >0.5	$\beta \gtrapprox 0.5$
Child selection	-	$\beta \leq 1$	$\beta \leq 0.5$	$\beta \leq 0.5$

(a) Parent score is 0.

Orig. score	P=C1=C2	P>C1,C2	C1≥P>C2	C1>P>C2	C1,C2>P
Parent	β >0	$\beta \geq 0$	β >0.5	β >0.5	$\beta \gtrapprox \frac{1}{3}$
Child	β=0	-	$\beta \leq 0.5$	$\beta \leq 0.5$	$\beta \lessapprox \frac{1}{3}$

(b) Parent score is non-zero.

Table 3: β values resulting in selection of parent (P) and children (C1, C2) for a 2-level binary tree.



(a) 2 Level

(b) 3 Level

Figure 6: Annotated scenarios where the nodes along the ground truth path are shaded.

the summarization for various original score configurations and values of β, and verify that for a given range of β values the outcome of SUM meets intuitive expectations.

2-level tree. In the first constrained scenario we consider a 2-level tree, as shown in Figure 6a that consists of a parent node with two children. The parent node has two configurations: (*i*) zero original score and (*ii*) non-zero original score. We considered every relevant combination of relations between the original scores of the nodes in both cases, as shown in Tables 3a and 3b, respectively. We then devise the range of β values for each configuration, causing SUM to pick the parent node (second row of each table) or one of the child nodes (third row of each table). The gray cells of Tables 3a and 3b represent the expected outcome for each configuration.

With both original score configurations, we were able to limit the number of possible outcomes since SUM bases its decisions exclusively on scores associated to nodes, independent of the identity of siblings. From both tables we can conclude that a low β value is necessary for the selection of the right node. The selection of the parent node is more sensitive than the selection of its children, since selection of the parent hinges on not only its own score, but also those of its children. Our evaluation shows that selection of the parent by SUM requires a β value that is moderately large. Furthermore, the selection of the parent requires that the children are relatively equal in score or that the score of the parent is sufficiently large, which is reasonable.

3-level tree. We now consider a slightly more complex scenario with a 3-level tree depicted in Figure 6b. Given that SUM does not differentiate based on the identity of sibling nodes, there are only three distinct outcomes of node choices, represented by the gray nodes in Figure 6b. Scenarios that have any other node as representative are equivalent to one of these three with an exchange of original scores between sibling nodes. In the following we analyze the possible scenarios that can lead to the choice of each of the gray nodes in Figure 6b.

1. *Root*: The root has a high likelihood of selection only if an original score is associated to it and β is set to a high value. Relatively balanced original scores at children nodes also should lead to the selection of the root, which SUM performs when β is sufficiently high. In general, high β is necessary for SUM to select the root so that a sufficient amount of score from its descendants is propagated to the root node in spite of damping.
2. *Intermediate Non-Leaf*: The selection of the intermediate node is sensitive to the scores of its children, its parent, and the scores of the subtrees of the siblings. A low value of β allows the score of the intermediate non-leaf node to be kept greater than its children, but also to prevent a sufficiently high score from being aggregated at its parent. The scores at the children of the intermediate non-leaf node have a small impact at the root to repeated damping.
3. *Leaf*[1]: Selection of the leaf node requires imbalance in the score distribution. Scores that are balanced at any point in the tree at or above the leaf node in the tree indicate that an ancestor should be chosen as the most representative node. β must not be too large in order to ensure the selection of leaf nodes. Otherwise, there is a higher likelihood of selection of an ancestor notwithstanding the score imbalance in the leaf nodes.

   Our analysis above shows proper tuning of β ensures that SUM accurately selects the right nodes in these limited scenarios. We believe that the basic properties displayed in these constrained scenarios will also be maintained when operating on much larger trees. Given that selection of the root is generally undesirable due to its lack of informative value (being it the most general node), a relatively low value for β is preferable as it favors intermediate nodes.

Selection of β value. Our previous analysis on synthetic ground truth motivates the selection of an optimal β value. Clearly, neither our 2-level case nor our expanded 3-level case provide a definitive β value. The 3-node case motivates a β value of approximately 0.5, or more generally $\frac{1}{|c|}$ where $|c|$ is the number of children, but that is only when a child node's score is sufficiently larger than its parent's score. The 3-level case does not consider specific values of β, yet it does underscore that multi-level propagation must be considered since a greater number of scores accumulate as one moves up the tree. Furthermore, we must also consider that the number of children a category has

[1] An intermediate node with children may be considered a leaf if all of its descendants have zero score. See the subtree on the right-hand side in Figure 6b for an example.

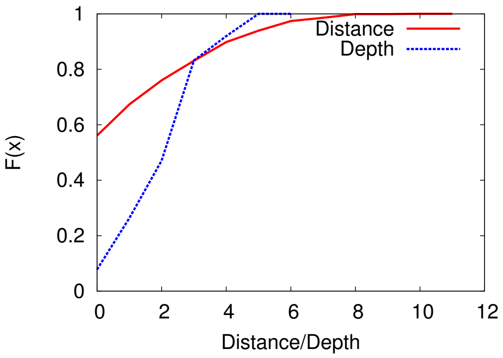


Figure 7: CDF of the distance and depth stability of SUM output.

is unrestricted, and in the category tree we utilize, less than half of the categories have two or fewer subcategories. We leverage the fact that approximately three-quarters of the categories have at most four subcategories, and that intermediate non-leaf nodes may have a non-zero score, to motivate a $\beta = 0.25$. A β value of 0.25 requires that a parent has a sufficiently large score, although not necessarily larger, compared to that of its children in order to be chosen as the summary. We utilize $\beta = 0.25$ since it favors selection of subcategories unless the subcategory scores are approximately equal to each other or the parent has a sufficiently large score of its own.

4.3. Evaluation of SUM Properties on a Real Dataset

We focus on *stability* and the *quality* of the summaries SUM generates when operating on the real web activity trace described in Section 4.1.1. In performing such an evaluation, we also investigate the effects of the algorithm parameters on the semantics of summaries.

Stability of Summarization. We assume that the most significant interest points of users (*i.e.*, summaries of interests) would largely be the same over the course of five days, even if the users may not visit exactly the same websites over time. We run our algorithm in multiple rounds where each round consists of randomly splitting each user's activity in two and applying our algorithm on each half. The information we obtain from our analysis is the distance between the two categories output by our algorithm, one category for each half of the data, as well as the depth of their lowest common category. Our results are summarized in Figure 7 which shows CDFs for both the distance and depth measurements.

Distance between two summaries. The stability in the selected category is denoted by hop distance in the tree between the two summaries from the two activity sets. A small distance indicates that the selected category remains focused on a particular area of the tree as well as a particular level of specificity. Large distances signify that there is a high degree of instability in our algorithm's summarization process when given different samples of data for the same user.

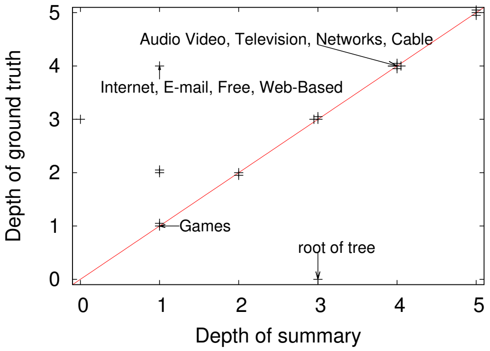


Figure 8: Scatter plot of summary depth vs. ground truth depth for individual users. The line marks summaries that match the depth of ground truth node. Summaries above the line are too specific whereas those below the line are too general. Individual users have concentrated activities resulting in quality summaries.

As Figure 7 demonstrates, large distances are uncommon with approximately 10-15% of the summaries having a distance greater than or equal to five. Large distances, as we will discuss further, are a consequence of both balanced categorizations and our algorithm's ability for extracting specific summaries. High stability has a high likelihood as indicated by 58% of our summarizations exactly matching and 80% of our summarizations having a distance of at most three. In particular, our algorithm has an average distance of 1.37, in comparison to 8.58 if two nodes are randomly chosen, which reinforces that summarizations are kept in a particular locality in the tree.

Depth between two summaries. The degree of specificity present in stability analysis is denoted by our measurement of depth, or most specific common category. The degree of specificity is inexorably correlated to the distance between the categories, given that the greater the distance the higher the probability that the root is in the path. Our stability analysis demonstrates that the root is actually unlikely to be along the path, and approximately 10% of the summarizations have the root in common. The remaining 90% are of a more specific category with approximately 60% being more specific than a top level category. Specificity beyond the top level categories validates that even with instability present, the likelihood of remaining in a specific subtree is high. Furthermore, even when the top level category remains the same, our approach will continue to extract results of high specificity within the appropriate category. The stability of our summarizations are particularly good with an average depth of 2.42, in comparison to 0.14 if two nodes are randomly chosen, which indicates further that the results are kept within specific subtrees.

Quality of Summarization. We conduct a qualitative study of our algorithm's output on a subset of our high-activity users. We inserted all of a user's web activities into the category tree and manually analyzed the results. Manual inspection of the results enabled us to gain intimate knowledge of a user's categorized web activity and to better determine the utility and accuracy of said results.

The web activity of users, as we discovered, was skewed towards only a few of their activities, and SUM is able to handle users with skewed interests effectively. As long as the web activity was concentrated far from the root, which it was, then our algorithm sought out more specific categories under a path of categories specified by our choice score. The affinity that our algorithm exhibits for specific categories is shown as there are multiple specific categories on the line in Figure 8. The category “Games” is not a by-product of our algorithm’s affinity for specificity, instead it is the result of how web activity is categorized, which will be described later.

Summaries that are too specific may be present even under minor skew or balanced interests. The interplay between the damping factor and choice score in our algorithm may result in the selection of a correct category, despite a more general category being more appropriate. An example of a category that is more specific than necessary, albeit correct, is the category marked above the line in Figure 8 when the category *Internet* would have been sufficient. Note that *Internet* appears general but the activity in question was spread across search engines, e-mail, and other Internet-based interests more-or-less equally.

Results that are more general than necessary can result in information loss, although this is not always the case. A summary that is general may expose little about the actual interests of a user. In the worst case we simply get the root of the tree which either says nothing or that the user’s interests are balanced across many categories. In certain circumstances the general category is the result of mappings that are unable to be made more specific such as the selection of *Art*, not shown in Figure 8. Listing *Games* as just right may appear contradictory considering that *Art* is too general, but while both *Games* and *Art* had a large number of mappings, there remained to be insightful mappings more specific to *Games* whereas there were none more specific than *Art*.

4.4. Comparison Against Hierarchical Heavy Hitter (HHH) Detection

We further evaluate SUM with a case study of user-generated web activity. We will examine the web activity of a collection of users in order to discover what is common among them. This case study thereby demonstrates SUM’s effectiveness in summarizing the commonality of multiple users, not just summarization of a single user. We leveraged SUM’s flexibility, denoted by the generality of its definition, by utilizing multiple functions, each of which optimizations for different properties of our dataset.

We also use this section to compare SUM against a method aimed to detect Hierarchical Heavy Hitters (HHH). We use the following definition of HHH detection from [2] for our study:

> Given a category c at level i in the hierarchy, define $F(c)$ as $\sum f(e) : e \in subcategories(c) \wedge e \notin (\cup_{l=0}^{i-1} HHH_l)$. HHH_i is the set of HHH at level i, that is, the set $\{c|F(p) \geq \lfloor \phi N \rfloor\}$. The set of $HHH = \cup_{i=0}^{h} HHH_i$.

Note that by the definition of HHH, that there can be $\leq 1/\phi$ HHHs in total. We extend the HHH definition to allow for a function to be applied over the elements such that the result of $F(c)$ is not necessarily a straight sum, e.g. define elements as unique users instead of the aggregate activity for a category. The extended flexibility of what is defined as a HHH allows us to more closely compare HHHs to SUM.

4.4.1. Parameter Tunability

In this particular case study we demonstrate SUM's tunability in producing summaries of varying generality that represent 100 randomly selected users. Tunability, particularly variations in β, is an underlying feature of SUM since it allows for the importance of generality or specificity to be tuned as needed given fixed definitions for choice and stop score functions. (We forgo a discussion of additional groups of randomly selected users since we found many of the groups have results that coincide with our findings in Section 4.4.2.) We also showcase the ability of our approach in ignoring less important categories and selecting categories that are more informative.

We used a choice score function that is the inverse sum of web activity across all users for a given category. Our definition of choice score function favors subtrees with a large amount of web activity spread across a large number of categories. The stop score was simply the sum of web activity across all users for a given category. Intuitively, our definition of stop score function with this particular choice score function is tunable based on balance across subcategories. Our results support this claim of balance: Given $\beta = 0.2$, we produce a summary with the category *United States* which had web activity spread across a large number of subcategories; if we then tune β such that $\beta = 0.1$, we now get much more specific categories that include specific information about states such as Maine and Michigan. (Increasing β is not valuable since $\beta = 0.3$ produces the root, of which no useful information is garnered.)

The summaries produced with the functions above ignore more populous activities, of which we now explore for our focus on specificity. We define stop score and choice score functions as the number of users that have any amount of web activity in a given category. SUM produced a summary that includes *Yahoo Inc.* when given $\beta = 0.3$. Increasing β to 0.4 proves useful in increasing the generality of our summary to include *Companies* which is under the *Computers* category. The category of *Computers* was chosen as the more specific category, despite large amounts of activity under the category *Internet*, due to more diverse activity in the former category. In fact, we lost information about Internet usage by choosing the category *Computers* yet we did gain more specific information about this more popular category.

Tuning the parameters for the summarization process changes the salience of the resulting summaries, as we have shown. Our two instances of summarization given in the previous two paragraphs had a difference of 0.2 in their β parameter for equivalent summaries, e.g. $\beta = 0.3$ and $\beta = 0.5$, and the two respective summaries provided no useful results, i.e. the root category. The common attribute between the two summarization cases above stems from the relatively

Choice Function	**SUM Category**	**Most specific HHH** ($\phi = 0.1$)	**Most specific HHH** ($\phi = 0.05$)
Summation of activity	Internet, Search, Google	Internet, Search, Google	Audio_Video, Television, Networks, Cable
Number of users with the activity	Computers, Companies, Yahoo_Inc.	Internet Computers	Regional, North_America, United_States
Logarithm of the summation of the activity	Business, Clothing, Footwear, Consumer_Goods_&_Services	Regional, North_America	Regional, North_America, United_States

Table 4: Effects of different choice functions.

low β parameters, as β approaches 0.5 the likelihood of producing the root increases regardless of the functions used.

The tunability of HHH detection is directly in contrast to that of SUM. The extended HHH allows for both ϕ and the HHH function to be modified but the degree of tunability remains limited. The parameter of ϕ is similar to the combination of β and our stop function since ϕ indirectly defines how specific the HHHs may be. We found that values of $\phi = 0.1$, produce only general results with a depth of 1 or 2 at most for our entire set of high activity users. The number of categories defined as HHH as tractible with only 4 HHHs, of which the root is always a HHH. Decreasing ϕ to 0.05, and therefore increasing the specificity, produces many more specific results with depths of 3 and 4 not being uncommon. Unfortunately, the number of categories that qualify as HHHs increases to 11, many of which still have a depth of around 1. Furthermore, many of the categories that qualify as HHH with $\phi = 0.1$ also qualify with $\phi = 0.05$. While HHH detection is tunable, the degree to which it can be tuned is limited.

4.4.2. Exploratory Choices

In this case study we collected summaries that utilize information across all of our high-activity users. We will focus primarily on functions we used for choice scores. (We found that for this particular set of users, changes in the stop score function generally resulted in no significant change in specificity.) Furthermore, in all cases we used $\beta = 0.2$ to keep our results meaningful. We used the values of $\phi = 0.1$ and $\phi = 0.05$ for HHH detection and selected only the most specific HHH as the summary. The choice score functions that we used were the same as those provided in Section 3.2, namely: summation of activity, number of users with the activity, logarithm of the summation of activity.

Choice: Summation of activity: Summation of activity provides insight into activities that are performed heavily by users. This approach confirmed our insights from our inspection of individual users, particularly that *Google* search is the most heavily used activity among the users. Clearly, Google search is a popular Internet tool used by millions of users. The interesting information we were able to glean from this is that all the other activities across all users were skewed toward Google search.

Search, particularly Google, is incredibly popular. In fact, users frequent it multiple times over a short period of time, more than any other activity. We suspect that many of the other activities frequented by users are tied directly to search rather than a direct URL. This summary is thereby a confirmation of the role that search plays in people's browsing habits.

Summaries from HHH detection disagree on whether or not *Google* search is a good summary. For $\phi = 0.1$ the most specific HHH is identical to SUM's summary confirming that it is the most significant category by web activity for the given choice function. For $\phi = 0.05$ the most specific category is instead network cable under the audio/video category. Audio/Video is an appropriate category when given our set of high activity users. The drawback of this summary is that it has the side-effect of a more specific category, audio/video, over-shadowing a more popular category, search, since HHH approaches examine the entire tree without considering semantics.

Choice: Number of users with the activity: The number of users with each activity allows us to ignore bias completely and look solely at popularity. This approach provided us with *Yahoo Inc.*, a large technology company, as our summarization. Recall that Yahoo provides a variety of services for users such as e-mail, news, and search. This summary has similar properties of the previous summarization, specifically that the services provided are broad enough that we cannot definitively determine the end interest. Despite this limitation, there are still insights we might glean from this summarization.

The Yahoo summary provides an unexpected insight that the total number of users are more interested in visiting Yahoo instead of Google. In particular, it demonstrates that there is a large degree of skew when we solely look at activity, as in the case of using summation of activity to result in Google. A large user base that provides little traffic would lead to such discrepancies. This approach clearly indicates that Yahoo provides enough services directly to end-users that it is able to pick up more distinct traffic from different users than Google.

Results from HHH detection diverge from SUM's output when given the number of users with the activity as the metric. For $\phi = 0.1$ we get both Internet and computers but both categories are top level categories and therefore are general summaries. Furthermore, the category of computers is a more general category of our summary which indicates that our summary is within an appropriate subcategory, the difference is that our summary produces a category of arguably higher utility. We get better specificity when given $\phi = 0.05$ which produces the United States region as its summary. We found through manual inspection that the United States region has a moderate amount of activity but SUM hones in on hotstops of activity. HHH is therefore able to uncover categories that SUM overlooks but HHH remains to have the limitation that if only the most specific category is examined, there are only a few compared to the entire set of HHH, then the more significant categories are unnoticed.

Choice: Logarithm of the summation of activity: The logarithm of the summation of activity is a heavily damped choice function that primarily seeks unbroken chains of categorizations with mapped activity. This approach provided us with the category of *footwear*, which is listed under clothing and more generally business. The use of a choice function that is heavily damped brings forward information such as this that might otherwise be considered a niche or unimportant. This selection of category is further highlighted when we utilized a different stop score function, namely the maximum of the activity, which summarized more generally to the category of business. Business is therefore an important category that further highlights a tangible interest of users.

The importance of footwear in our summarization stems from the fact that shoes, in comparison to automobiles, are a relatively low cost item. Shoes are purchased more frequently and it is common that people have many pairs of shoes. It is then made apparent that active research, albeit in fewer numbers, revolves around shoes. Considering that a stop score function provided us with a different level of specificity , shoes are not the sole interest of users. The discrepancy in the level of specificity can further prompt investigation, if desired.

Results from HHH detection completely disagree with SUM's output for this particular choice function. For both $\phi = 0.1$ and $\phi = 0.05$ we are given the North America region as our summary except $\phi = 0.05$ also provides the United States as a more specific region. The fact that both values of ϕ produce similar summaries indicates that this category is likely to be significant. This does not invalidate SUM's output but instead highlights further the differences between a bottom-up and a top-down approach of HHH and SUM, respectively.

Concluding insights. Our approach for summarization provided a means to effectively learn about an individual user, and for this case study a group of users. We summarize the results of our case study in Table 4. As we have discovered, the most popular activity among the users is Google search for indirectly providing access to information whereas more users appear to consume information directly from Yahoo. When we focus our choices more on general mappings we obtain activity for footwear. This last finding is particularly interesting since it demonstrates that activity in the shopping category, of which there are tangible items, have more general categorizations than categories with much higher activity.

We have also found that HHH detection and SUM produce complementary results. HHH approach and SUM typically produced summaries that differed in category, many times the differences were substantial. The summaries produced by HHH detection focus on categories with activity at lower depths in the tree due to its bottom-up approach. The bottom up approach makes HHH summaries prone to localization which is demonstrated by the repeated summaries involving the North America region. While the more general summaries produced by HHH detection typically agreed with those produced by SUM, SUM was able to have greater flexibility than HHH since the choice and stop functions separates the selection of branch and generality where as parameters of HHH detection does not.

4.5. Limitations

SUM has been shown effective and relatively stable. The summaries it produces change only as a result of a significant change in the input data. This is the case even though the summaries are not overly general, which is what we have shown happening with HHH. On the other hand, our approach is not without its limitations. In particular, our approach is sensitive to the quality of categorization and to the functions that are defined for the stop score and choice score. Ill-defined functions, as we will show, may provide results that appear significant and logical but are in fact misleading.

The choice function of the average activity across all users in a particular category would appear to emphasize an *average* user. The average in this case aims to avoid extremes caused by a large number of users with activity in a particular category. We applied this approach on our data and car sales was given as a summary. We suspected that our group of users, on average, were interested in car sales when there was only one such user with an inordinate amount of activity present in that category. This is just one such example of a poorly defined function and the ease at which they may be defined in general.

The categorizations provided as input into our algorithm also serve as a potential limitation. The damping process that is utilized for the stop scores, and consequently the degree of specificity, is affected by the specificity of the categorizations. We alluded to such limitations during our discussion of ground truth in which more general categorizations can overshadow the more specific categorizations. Recall that a category's score is its own original score plus a damped sum of its subcategories. Given a large enough original score, regardless of the importance of a subcategory's score, the more specific category will never be chosen. The weakest link in our process then becomes the categorization process itself, which is another problem entirely, one that we do not attempt to solve in this paper.

5. Related Work

The quantification of the importance of information defined by a graph-based structure has seen significant activity in the research community, particularly with respect to web search. Aimed to identify the most significant vertex in a graph, graph centrality algorithms were first proposed in the context of social network analysis. The algorithms were soon expanded to a wide variety of large-scale graph analyses including online social networks, urban networks, and infectious disease spreading networks. Different from simplistic measures which only consider shortest paths or direct neighbors, path ensemble-based graph centrality measures such as PageRank [1], HITS [10], and the Katz [5], are shown to be more effective in providing a global view of importance as they take into account all nodes connected to the nodes in consideration. PageRank (or weighted PageRank [11]) recursively calculates the importance the node receives and loses through its edges. While PageRank and HITS works well on general graphs, they work suboptimally with tree structured graphs such as ours as they aggregate the majority of the importance in the root node which also happens to carry the least amount of information. The Katz metric for a node is determined by computing the influence between nodes over every possible path where an attenuation factor determines the exponentially fast rate at which influence is damped as path length increases.

Inspired by the fact that graph centrality is designed to work well in multihop, multidimensional topologies, we cast a variant of the Katz algorithm to the new domain of discovering the most significant node (*i.e.*, central node) in a hierarchical tree encoding of Internet activity logs. Direct application of Katz, however, suffers from imbalances when a tree has varying breadth and makes its output have regions of high and low importance. We overcome these shortcoming by developing two scoring mechanisms in which the metric individually considers sibling (breadth) and descendant (depth) relations of nodes.

In database research, multidimensional aggregation (or multidimensional summarization) has been a topic with a rich history. A number of methods were proposed for summarizing multidimensional data. [12, 13] described methods summarizing data by comparing pairs of data points. However, these non-hierarchical, flat summarizations do not capture parental relationships embedded in many of the multidimensional data. [14] proposed a hierarchy-aware summarization method that takes into account hierarchical properties. The Minimum Description Length (MDL) aggregates importance of significant data points in its summarization.

The concept of multidimensional aggregation was borrowed in the computer networking field and greatly advanced. Specifically, the concept was used in discovering Heavy Hitter (HH) IP addresses (equivalent to data points in DB research), i.e., hosts responsible for generating a large amount of network traffic. Similarly to the change in the DB research, networking research also moved its focus from flat structure to hierarchical structure. A Hierarchical Heavy Hitter (HHH) is a HH in a hierarchical network. In an IP network, the hierarchy is often represented as a *trie* of IP addresses. The study on HHH further advanced the field by providing a variety of algorithms for finding hierarchically significant regions whose subregions are not significant by themselves (*i.e.*, the significance of the region is only discoverable by aggregating significance of subregions) [15], which is equivalent to finding the most representative node in a hierarchy for the purpose of summarization. The study on HHH also improved multidimensional aggregation in terms of computational complexity by developing partially updatable, online algorithms [2, 3, 4, 16].

In our trees, where ancestors are only semantically super-ordinate to their descendants, the methods of HHH cannot be applied as they strictly add up all data points from bottom up. In our algorithm, we employ parameters to tune goodness of the summary by adjusting the balance between specificity and generality.

Ontology Summarization (OS) approaches the problem by considering semantical view from ontology [17, 18, 19, 20]. They elect the most representative node from ontology graphs that are structured based on the meaning (semantics) of the relationships between data. Different methods of OS develop various metrics to measure importance of words in graph nodes as well as to aggregate them in to a summary. [17] develops methods to weigh simple words versus compound words, [18] applies a text summarization approach to score nodes for summarization. While these approaches score *semantic* importance of nodes using specific grammatical structure dependent to languages, we employ a graph centrality metric that does not require semantic understanding of the terms comprising the ontology.

6. Conclusion

In this paper we presented SUM, an algorithm for summarizing hierarchically categorized data. SUM is applied to the network activities of a user mapped on a category tree of a hierarchy to determine which node best represents and summarizes the interest of the user. Such a node needs to be as specific as possible, but not too specific such that it is no longer characteristic of the user's activities. SUM can successfully find such a node by employing two node-scoring mechanisms: one method that assigns the "choice score" to nodes of the tree which will help find a node that is as representative as possible in terms of characterizing the user's activities, and another method that assigns the "stop score" to nodes in order to prevent the summary from being too specific, i.e., to go too deep in the selection of the most representative node. After first assigning every node in the category tree an original score that reflects the magnitude of the importance of the user's data mapped to this node (such as the frequency of visiting a web site related to this node), SUM uses the score propagation mechanisms to assign and propagate choice scores and stop scores throughout the tree. At that point, the tree can be traversed using the scores to reach the most specific yet representative node, i.e., the summarization of the user's network activities.

Evaluating our algorithm on a dataset from a large ISP, we demonstrated that our algorithm has desirable stability properties when applied to web activity. We were also able to determine that the results had a specificity that matched the distribution present in web activities of individual users as well as groups of users. In our comparative study of SUM against the Hierarchical Heavy Hitter (HHH) detection approach, the examination of the summaries produced by the two approaches showcased the flexibility of data exploration SUM offers thanks to the two score functions it uses.

7. References

[1] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web.

[2] G. Cormode, F. Korn, S. Muthukrishnan, D. Srivastava, Finding hierarchical heavy hitters in data streams, in: VLDB, 2003.

[3] G. Cormode, F. Korn, S. Muthukrishnan, D. Srivastava, Diamond in the rough: Finding hierarchical heavy hitters in multi-dimensional data, in: ACM SIGMOD, 2004.

[4] Y. Zhang, S. Singh, S. Sen, N. Duffield, C. Lund, Online Identification of Hierarchical Heavy Hitters: Algorithms, Evaluation, and Applications, in: ACM SIGCOMM, 2004.

[5] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.

[6] M. Newman, Networks: an introduction, Oxford University Press, 2010.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 2nd Edition, MIT Press and McGraw-Hill, 2001.

[8] A. I. Inc. (August 2013). [link].
URL `http://www.alexa.com/topsites/category`

[9] S. Khemmarat, S. Saha, H. H. Song, M. Baldi, L. Gao, On understanding user interests through heterogeneous data sources, in: PAM, 2014, pp. 272–274.

[10] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (5) (1999) 604–632.
URL `http://doi.acm.org/10.1145/324133.324140`

[11] W. Xing, A. Ghorbani, Weighted pagerank algorithm, in: Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on, IEEE, 2004, pp. 305–314.

[12] N. Thaper, S. Guha, P. Indyk, N. Koudas, Dynamic multidimensional histograms, in: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, SIGMOD '02, ACM, New York, NY, USA, 2002, pp. 428–439. doi:10.1145/564691.564741.
URL `http://doi.acm.org/10.1145/564691.564741`

[13] S. Guha, N. Koudas, K. Shim, Data-streams and histograms, in: Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, STOC '01, ACM, New York, NY, USA, 2001, pp. 471–475. doi:10.1145/380752.380841.
URL `http://doi.acm.org/10.1145/380752.380841`

[14] L. V. S. Lakshmanan, R. T. Ng, C. X. Wang, X. Zhou, T. J. Johnson, The generalized mdl approach for summarization, in: Proceedings of the 28th International Conference on Very Large Data Bases, VLDB '02, VLDB Endowment, 2002, pp. 766–777.
URL `http://dl.acm.org/citation.cfm?id=1287369.1287435`

[15] C. Estan, S. Savage, G. Varghese, Automatically inferring patterns of resource consumption in network traffic, in: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '03, ACM, New York, NY, USA, 2003, pp. 137–148. doi:10.1145/863955.863972.
URL `http://doi.acm.org/10.1145/863955.863972`

[16] G. Cormode, F. Korn, S. Muthukrishnan, D. Srivastava, Finding hierarchical heavy hitters in streaming data, ACM Trans. Knowl. Discov. Data 1 (4) (2008) 2:1–2:48. doi:10.1145/1324172.1324174.
URL `http://doi.acm.org/10.1145/1324172.1324174`

[17] S. Peroni, E. Motta, M. dequin, Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures, in: The Semantic Web, Springer, 2008.

[18] X. Zhang, G. Cheng, Y. Qu, Ontology summarization based on rdf sentence graph, in: Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 707–716.

[19] X. Zhang, G. Cheng, W.-Y. Ge, Y.-Z. Qu, Summarizing vocabularies in the global semantic web, Journal of Computer Science and Technology 24 (1) (2009) 165–174.

[20] C. E. Pires, P. Sousa, Z. Kedad, A. C. Salgado, Summarizing ontology-based schemas in pdms, in: IEEE ICDEW, 2010.