Toward the Most Representative Summaries of Network User Activities

Joshua Stein[§], Han Hee Song[†], Mario Baldi[†], Jun Li[§]
[§] University of Oregon, [†] Narus Inc.
{jgs,lijun}@cs.uoregon.edu,{hsong,mbaldi}@narus.com

***Abstract*—A summary of a user's Internet activities, such as web visits, can closely reflect their interests and preferences. However, automating the summarization process is not trivial as it 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, we introduce two scoring mechanisms that cooperatively optimize for polarizing criteria. Having mapped user activity information onto a category tree, the scoring mechanisms highlight the most representative tree node; the node provides an aggregated view, i.e., a summary, of the activities most representative of the user. We evaluate our approach by summarizing web activity on the network of a large cellular service provider to devise interests of individual users as well as user groups.

I. Introduction

As people become connected via the Internet more than ever before, they also transfer a substantial amount of information about themselves. Such information is embedded, explicitly and implicitly, in their network traffic; examples include web site visits, data exchanged using different (mobile) applications, and GPS coordinates sent from their mobile devices. While these data provide an opportunity to discover rich information about the users, it is difficult to consume because it encompasses an extremely large number of very detailed and diverse data items.

Summarization is hence required to make such information manageable and practically usable. The difficulty in summarizing is striking the right balance between generality (which implies loss of information, i.e., detail) and specificity (which entails dealing with a large amount of information), for which there is no golden rule. The research on finding hierarchical heavy hitters make significant progress in summarizing heavy hitter nodes in network topologies [1], [2], [3]. However, as their target applications are limited to IP prefix-based *trie* structure where ancestor-descendant relationships are strictly enforced by their IP addresses, their algorithms cannot be directly applied to our less structured context where ancestors are only *semantically* super-ordinate to their descendants.

To make disparate pieces of user information, such as network activities users conduct and preferences they express in their on-line profiles, comparable, we propose **SUM (Summarizer for User inforMation)**. SUM operates on a categorization of such user information onto a single category hierarchy. Once user information, e.g., web activity, is standardized this way, SUM searches for the most representative tree node, which provides a summary of the user information within the hierarchy. In this paper, activity of web service visitations is used as a sample type of user information.

SUM devises two scoring methods based on Graph Centrality [4] to perform a search on the category hierarchy: *choice score* and *stop score*. Beginning from the root of the tree, for each node, we assign a choice score which determines a traversal direction, i.e., which child node to consider when traversing the tree. At the same time, we assign a stop score which determines traversal depth by choosing a sweet spot between general and specific node on the branch the choice score selects. 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 categorizations*, 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.

Contribution. To the best of our knowledge, this is the first work that systematically summarizes network user activities on a large scale. To this end, we design and implement the SUM algorithm that flexibly chooses a summary leveraging on the concepts of graph centrality, for a user (or a group of users). Our algorithm can be easily tuned for the summary to be more general or specific, depending on the application. Finally, we evaluate SUM using web browsing history collected from a large Cellular Service Provider (CSP) in North America covering 4 million web visits from 150,000 Internet users for five days. The summaries produced by SUM have high stability under vary amounts of user data (e.g., web service visits); the summaries only differ by 1.37 categories on average and have a depth, specificity, of 2.42 on average. Furthermore, the summaries produced by SUM are relevant even in the presence of skewed interests.

II. Background

In this section we discuss the components necessary for summarization in addition to the properties a summary should exhibit. The summarization process requires data, to which we

978-1-4799-4852-9/14/$31.00 2014 IEEE

(a) Flat Category (b) Tree

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

(a) Graph (b) Tree

Fig. 2: Explicit information of categorical semantics.

add annotations in the form of the category of an ontology the data may fall under.

A. Categorization Ontology

The ontology the summarization process operates on 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 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. This approach has the limitation that keywords do not relate to one another, which would enable us to derive strong connections between data items. 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). 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. 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 root category) 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). The presence of multiple paths introduces several challenges; in particular, it is necessary to know the path that was taken during the categorization process otherwise semantic information may be lost. To 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 that are appropriately placed within the tree force the meaning of categorized data to be explicit.

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

B. 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 II-A. 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 I 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.

C. 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. 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.

III. *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 III-A, the assignment of initial scores, dubbed original scores, to individual tree nodes in Section III-B, the propagation of the initial tree node scores throughout the tree into new scores in Section III-C, and then the summarization process in Section III-D.

A. 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 assumption described in Section II-B, 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 is 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.

B. Original Score Function G

The original score function G is a function that translates the data that are 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 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.

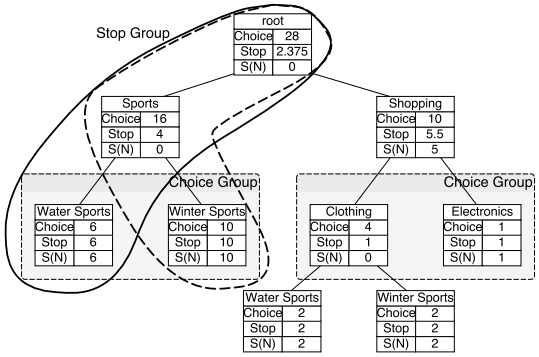


Fig. 3: Calculation of choice and stop scores given a single original score function. The scope of usage of the stop score is shown as being between a parent and child whereas the scope of usage of the choice score is shown as being between siblings. $\beta = 0.25$, $G =$ sum of activities.

C. 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.

1) Advantages over a single score: Propagating two scores separately helps quantify two incomparable properties (i.e., significance of the category among its siblings and the significance of the category compared to its subcategories) that are otherwise difficult to be aggregated into a single score. Alternatively, if using a single score to encode both properties, it will become infeasible to compare scores as we have lost information in the process. For instance, if we look at Figure 3 then we can see that if we utilized the stop score for our choice of category we would choose the "shopping" category which is less representative than the "sports" category. Conversely, if we utilized the choice score in Figure 3 then the score propagates independently from depth but the root would always be chosen since it represents all the data in the tree. A single score fails to represent the fraction of the score that is contributed by the

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 II: Definition of terms.

choice score or the stop score. Even after normalization shown in Section III-B, 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.

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 [5]. Among many implementations of the centrality measures, we use Katz centrality metric [4]. Different from simpler measures that only considers either a single path (e.g., shortest path [6]) or immediately neighboring nodes (e.g., common neighbor or Adamic-Adar measures [6]), 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 tree, 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 III-B. The score propagation function $F(i)$ is recursively defined as

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

where $G()$ is the original score function introduced in Section III-B. 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 IV.

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.

Algorithm 1 RepresentativeNodeSearch(node)

```
summary = node.category
maxStop = maximum subcategory stop score
maxChoices = set of subcategories w. maximum choice score
if node.stopScore < maxStop then
    summary = ∪_{c∈maxChoices} RepresentativeNodeSearch(c)
end if
return summary
```

3) Choice Score: The choice score is one of the two scores that is propagated during score propagation. For each branch in the tree, we assign the choice score such that a subtree closest to the summary (i.e., centroid) can be selected.

The motivation for the choice score is that categories are not guaranteed to have either equal depth nor are they guaranteed 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 $\beta \neq 1$.

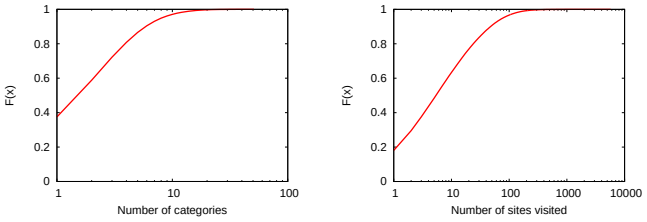
4) Stop Score: The stop score is the second of the two scores that is propagated during score propagation. The stop score, unlike the choice score, is not used to compare sibling subcategories against one another since the stop score is designed strictly for comparisons based on depth of the tree (i.e., between ancestors and descendent). 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.

D. Searching the Representative Node—Summarization

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 pseudo-code provided in Algorithm 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.



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

Fig. 4: Cumulative distributions of user web activity

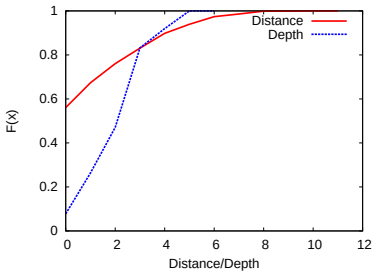


Fig. 5: CDF of the distance and depth stability of SUM output.

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 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.

IV. Evaluation

A. Evaluation Methodology

1) Datasets: **Category Tree.** The ontology we used for categorization is the same as the one utilized by Alexa [7] with only minor variations [8]. 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. The number of subcategories of a single category varies largely across the tree and in some instances a large number of subcategories result from degenerate cases, e.g. listing every country in a certain part of the world. 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 web site, is mapped on a category as specific 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 [8] 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 4a and 4b 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 web sites across at least 15 distinct categories.

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 and analysis. The choice score function and parameters are set such that the subtree with the most data will be chosen.

B. User Study

In this section we analyze the results of our summarization approach with real user data. We look at both the quality and the sensitivity, or stability, of the results.

1) 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 web sites over time. We run our algorithm 10 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 5 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

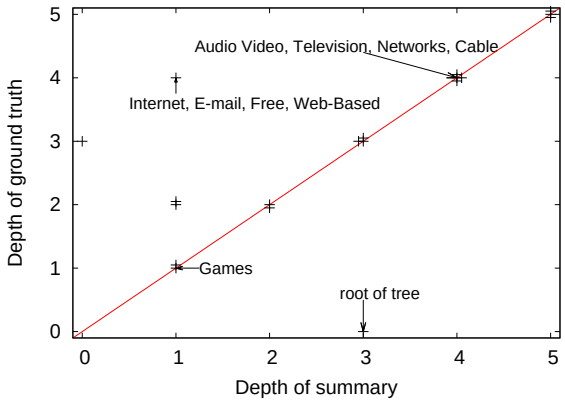


Fig. 6: Scatter plot of summary depth vs. ground truth depth for individual users. The $y = x$ line marks summaries that match the depth of ground truth node. Summaries above the line are too specific whereas those below the line are too general. Individual users have concentrated activities resulting in quality summaries.

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. As Figure 5 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 hops, 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 hops, in comparison to 0.14 if two nodes are randomly chosen, which indicates further that the results are kept within specific subtrees.

2) 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 $y = x$ line in Figure 6. 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 6 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 6. 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*.

REFERENCES

[1] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava, "Finding hierarchical heavy hitters in data streams," in *VLDB*, 2003.

[2] ——, "Diamond in the rough: Finding hierarchical heavy hitters in multi-dimensional data," in *ACM SIGMOD*, 2004.

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

[4] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

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

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

[7] "Alexa internet inc." April 2014. [Online]. Available: http://www.alexa.com/topsites/category

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