BUILDING A COLLECTIVE-EXPERIENCE ENGINE FOR EXPERIENCE-TRANSFER AMONGST WEB USERS

Jeremy Hall and Yasushi Kiyoki

Graduate School of Media and Governance, Keio University 5322,
Endo, Fujisawa, Kanagawa, Japan 252-8520

ABSTRACT

This paper describes the Collective Experience Engine (CEE), a system for indexing Experiential-Knowledge about Web knowledge-sources (websites), and performing relative-experience calculations between participants of the CEE. The CEE provides an in-browser interface to query the collective experience of others participating in the CEE. This interface accepts a list of URLs, to which the CEE adds additional information based on the Queryee's previously indexed Experiential-Knowledge. The core of the CEE is its Experiential-Context Conversation (ECConversation) functionality, whereby an collection of a person's Web Experiential-Knowledge can be utilized to allow a real-world conversation-like exchange of information to take place, including adjusting information-flow based on the Queryee's experiential background and knowledge, and providing additional experientially-related knowledge integrated into the answer from multiple selected 'experience donors'. A relative-experience calculation ensures that information is retrieved only from relative-experts, to ensure sufficient additional information exists, but that such information isn't too advanced for the Queryee to process. This paper gives an overview of the CEE, and the underlying algorithms and data structures, and describes a system architecture and implementation details.

KEYWORDS

Data Mining, Knowledge Representation, Collaboration, Web Application, Experiential Calculations

1. INTRODUCTION

The Web has had several evolutions of Knowledge-Expertise expression on the Web. Before the Web itself, people used bulletin board systems to gather and trade knowledge. The Usenet network was another such system. These systems and their successors such as Wikis, and Q&A domain-specific sites such as Stackoverflow provide effective ways of users pooling knowledge on specific, known subjects.

While current methods of finding information on the Web focus on ways to link keywords to explanation or discussion, it is still up to the user to find many potentially disparate sources of information, understand how they fit together, draw conclusions about the reliability and utility of various disparate information sources, and to do so with the potential handicap of not knowing the most efficient or proper keywords or wording for finding such sources.

DOI : 10.5121/ijdkp.2014.4101

This paper describes the Collective-Experience Engine (CEE), created to enable direct querying and visualization of the collective and untapped Experiential-Knowledge stored in the brains of all Web users (see *Figure 1*). Users of the CEE can learn from the experience of others, rather than having to build this experience on their own via extensive searching and surveying of websites. They can then make informed decisions about information sources on the Web, and have confidence in those decisions by relying on the knowledge and experience of other people participating in the system.

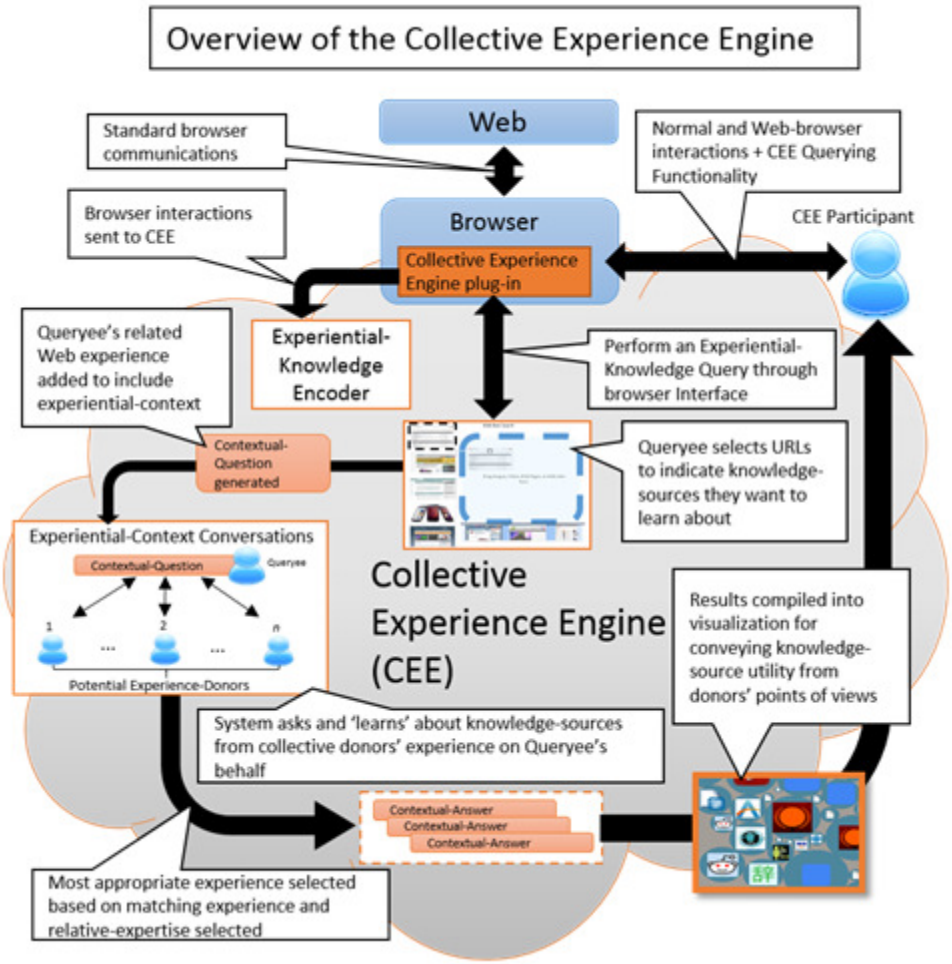

Figure 1. This figure gives an overview of the Collective Experience Engine, describing the data flow and processes within the CEE. A participant in the CEE uses their Web browser both for standard Web viewing, and for interacting with the CEE. While browsing, their interactions are sent to the CEE to be encoded as Experiential-Knowledge. When a Queryee submits a query, a special interface is used to select and then submit URLs describing the type of knowledge-sources they wish to learn about. The CEE inspects the Queryee's Experiential-Knowledge, and combines it with the explicitly selected URLs. Next, several 'Experiential-Context Conversations' occur between data-representations of the Queryee, and potential-donors of Experiential-Knowledge, who themselves are merely participants in the CEE. Once experience-donors have been selected, their Contextual-Answers are generated, which includes personal Experiential-Knowledge from

each donor, which was calculated to be related to the Queryee's original Contextual-Question. Finally the Contextual-Answers are returned as a custom visualization, presented to the Queryee in their Web browser.

The main feature of the CEE is the concept of Experiential-Context Conversations (ECConversations), whereby experiential-context negotiation occurs between proxies of people, rather than requiring explicit communications. An ECConversation is a model of a real-world interaction between people, where both communicating parties adjust the content of their communication as a function of their own, and the other person's experience and knowledge regarding the subject of their conversation. By modelling real-world Experiential-Knowledge transfer, we can automate and scale it to allow participants in the CEE to share their own and utilize others' Experiential-Knowledge.

Within the CEE, we explicitly define a model of individual experience, which is lacking in current knowledge discovery Web systems. With such a model, we can develop new visualization and computational functionality for creating new query methods. The CEE is functionally an overlay on top of existing Web-based knowledge-retrieval systems. The CEE reveals the utility and the context in which websites are useful, from the point of view of several people simultaneously. This allows people searching for information on an unfamiliar topic to make a decision about the quality and reliability of unfamiliar websites that they are viewing, without having to survey huge numbers of websites, or become experts in a topic themselves.

The CEE's ECConversation-algorithms determine how to combine the Experiential-Knowledge of each user. A query is a combination of a person's explicitly-selected URLs for describing the type of knowledge-sources they wish to understand, and a contextually-related portion of that person's captured Experiential-Knowledge. A person's Experiential-Knowledge is based on compiled Web-browsing behavior (not just viewed content, but actions in the browser itself such as tab-switching), and encodes the unique experience-signature of a person on the Web. The algorithms determine the appropriateness of which Experiential-Knowledge to include in the initial query, as not all experientially-related knowledge may be appropriate given a person's explicitly-selected URLs. The algorithms also determine who to select as Experiential-Knowledge donors from the rest of the participants in the system, and which additional Experiential-Knowledge from those donors is appropriate to include in the final results. Given two people that have chosen an identical set of URLs to submit as part of the initial query, due to the different experiences of those two people, and how the delta in Experiential-Knowledge affects the discovery of donors and which of those donors' Experiential-Knowledge to include, the results returned to two people who select identical initial query parameters are going to differ.

This paper gives an overview of the query process, and describes the ECConversation-algorithms in detail. A system architecture and prototype implementation is proposed, and the results-visualization process is described.

2. BACKGROUND: EVALUATING SOURCES OF KNOWLEDGE ON THE WEB

The ability to evaluate the reliability and potential utility of a website is greatly impacted by domain-knowledge regarding the website's topic. Low-knowledge browsers of websites depend on surface cues such as typology keywords highlighted by web-search engines, and thus fail to

interpret when a website is only tangentially related to their target knowledge [1]. People familiar with a topic use metacognitive functions to evaluate knowledge-sources. This means that in addition to the content and semantics of a website, other information such as the author, date, and the document type of the information is evaluated as well. Finally, past experience aids greatly in rapidly evaluating new knowledge-sources, as it enables people 'easily link prior knowledge to task requirements and to information found on the Web' [2]. Research into the difference in Web-research techniques between novices and experts shows that experts '[...]more often activate their prior knowledge[...]', and yet they '[...]show little differences in the way they search the internet...' [3]. This indicates that by providing people with more prior knowledge and understanding of the framework in which an expert evaluates their knowledge, we could rapidly improve the Web-search process for users unfamiliar with a topic they are searching.

P. Gerjets et al explains how searching on the Web has supplanted interaction with experts--for instance when diagnosing computer problems or seeking medical advice. The environment of the Web contains a large variety of complex information-domains, however the variability in terms of the quality and reliability of the information is substantial, possibly owning to a mix of both experts and laypeople providing information. 'As a result, Web users are required to appropriately evaluate diverse, potentially diffuse, or even contradictory sources of information[...]Web search is often related to personal decisions under uncertainty in domains characterized by fragile and conflicting evidence[...]it has been shown that searchers usually face difficulties in appropriately evaluating information during Web search' [4]. In order to overcome such search obstacles, users must put in a large amount of manual effort to understand and integrate knowledge about a search-topic, and perform a survey of websites to understand information variability and reliability.

In H. v. Oostendorp & S. R. Goldman, the importance of the formation of a mental 'Document Model', a mental model of the relationships between content and its sources, is described [5]. According to Bhavnani et al. [6], even the most common of topics on the Web is a 'fairly complex task', where people '[...]must first visit more than one general page to get an overview[...]and then visit specialized pages to get an in-depth understanding about specific concepts[...]'. The paper goes on to mention that because such '[...]search procedures are similar to what search experts have been observed to use, and because they are difficult to acquire just from using search engines like Google [...] motivate the design of new approaches to search systems that explicitly provide such guidance'.

In Britt and Aglinskas [7], they describe how 'Experts attend to many features of sources, and some, such as style, may be too subtle [for novices]'. Also, the lack of 'gatekeepers of credibility, such as editors and publishers' are placing greater emphasis on people's need 'to filter and evaluate information sources'. Experts are better able to deal with bias, as they use multiple criteria when evaluating sources, and not just base evaluations on content and the amount of information provided [8]. 'The fact that the amount of immediately available information is nearly unlimited on the WWW underlines the need for a reasonable selection of information[...] laypersons need to activate prior knowledge in order to integrate information from multiple texts and thereby build semantic connections between information from different sources. Finally, to gain knowledge about the sources, laypersons have to evaluate sources in terms of quality and credibility. This involves finding out about the author as well as his or her credentials, intentions, possible affiliations, and sponsors.' Bråten [9] also describes how experts overcome bias better than novices, especially when the content is written to be more easily understood, and thus gains a greater value than is necessarily warranted in the novice's eyes.

In developing the CEE, we have created a system to allow novices and people unfamiliar with a topic to understand how a relative-expert would collect information on said topic. Even if the novice doesn't understand the reasoning behind a relative-expert's choices, they get the benefit of the extra semantic, source, and other meta-cognitive knowledge that an expert has used to create their own mental 'Document Model' of related knowledge-sources. Not only should utilization of the CEE allow relative-novices to immediately gain confidence and the use of better-sources of knowledge, but by providing an overview of good usage of knowledge-sources for a given topic, the CEE gives people the ability to more accurately identify the reliability and usefulness of a website on their own.

3. OVERVIEW OF THE COLLECTIVE-EXPERIENCE ENGINE

Performing a query with the Collective-Experience Engine is an easy process, whereby a person defines a list of URLs to describe the type of information-sources they wish to better understand. Selecting URLs is made easy with an in-browser interface we've developed, where people can perform the add function from within a web browser window just by right-clicking on a page, or a link to a page whose URL they wish to add.

The CEE passively-collects Experiential-Knowledge, extracting the relations among URLs that that person has previously visited, and their utility to the user, in order to build a body of knowledge to use on behalf of the CEE participant. This passive collection occurs as a person utilizes a Web browser with a special browser plug-in installed.

When a query is submitted, the CEE first constructs a Contextual-Question, which is the Queryee's contribution to an ECConversation. The Contextual-Question not only encodes the initially selected list of URLs, but also includes contextually-relevant Experiential-Knowledge (see *Figure 2*). URLs based on relevant Experiential-Knowledge are chosen based on two factors:

- An experiential-relationship existing in the person's encoded Experiential-Knowledge from or to an explicitly selected URL.

- An appropriate relatedness-factor, calculated by comparing the relatedness of the explicitly-selected URLs to each other, and then to each experientially-related URL. This is meant to prevent contextually-unrelated experiential-relations from being included in the final Contextual-Question.

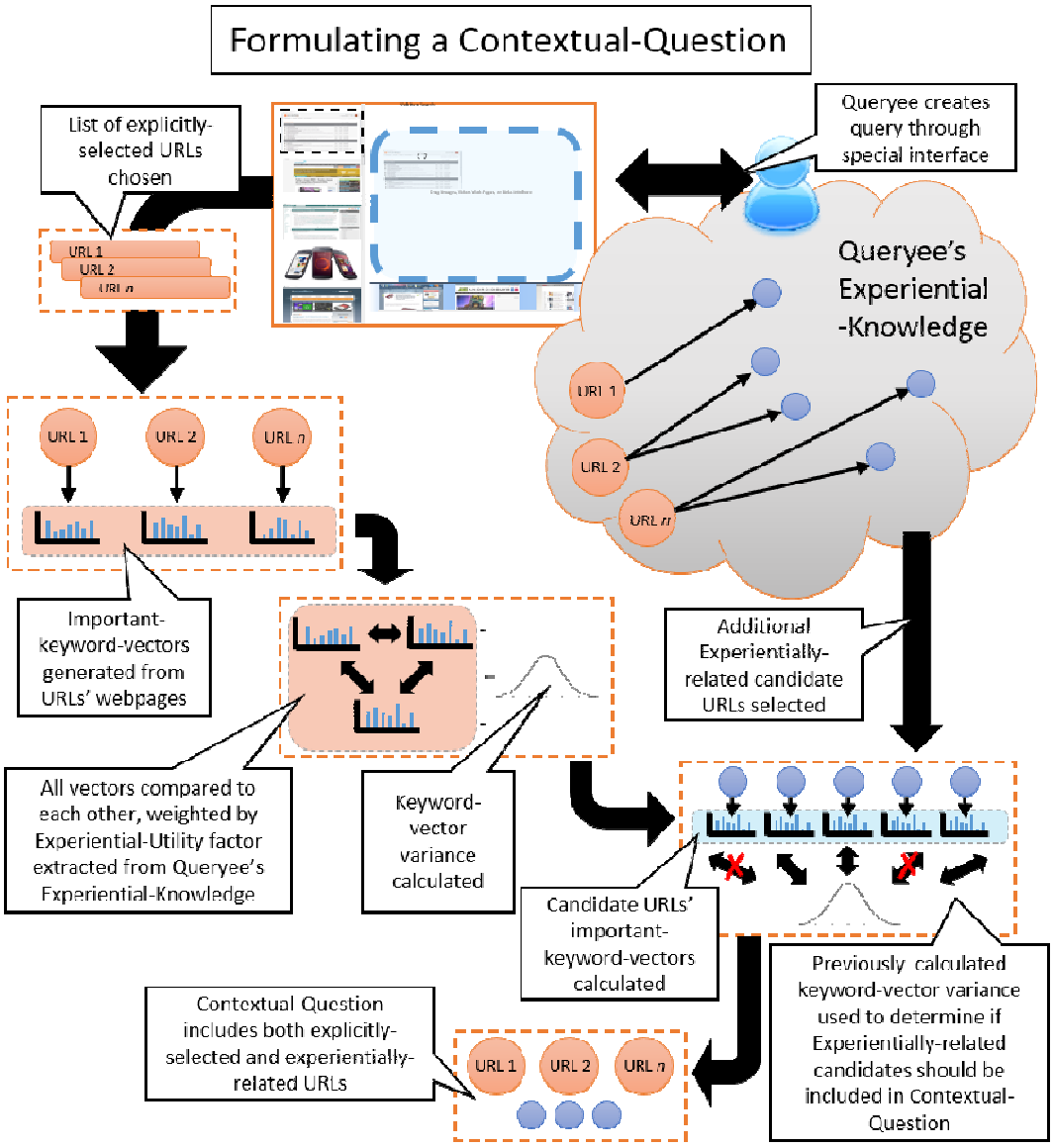

Figure 2. This figure describes in detail the process of formulating a Contextual-Question. The Contextual-Question is created as part of the query-process, and is used to find potential experience-donors, and then to describe the type of Experiential-Knowledge that should be donated to the Queryee. First, the Queryee explicitly selects URLs, which describe the type of knowledge-sources the Queryee wishes to learn more about. The CEE then extracts important keywords from each URL's website, calculates the variance pattern of the URLs' keywords, and then selects experientially-related URLs whose keywords are relevant within the same tolerance as the explicitly selected URLs'. The combination of explicitly-selected, and contextually experientially-related URLs make up a Contextual-Question

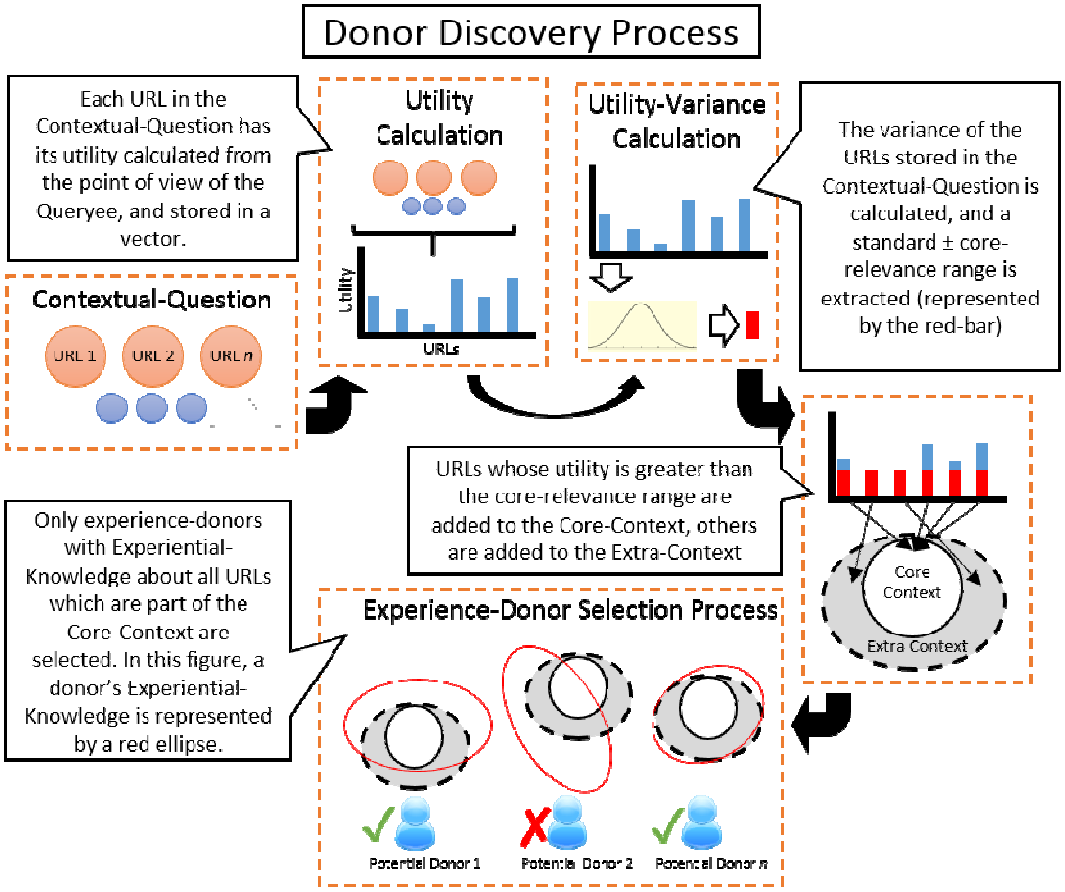

Figure 3. This figure describes the process of using the Contextual-Question to identify experiential-donor candidates. Any participant in the CEE is a potential donor of Experiential-Knowledge. A Contextual-Question first has the utility of each of its included URLs calculated, and the variance calculated from the resulting utility-vector. URLs whose utility is larger than a calculated variance-range are used as the Core-Context. All participants within the CEE whose Experiential-Knowledge contains the entire set of URLs within the calculated Core-Context become an potential experience-donor.

Next, Experiential-Knowledge donor-candidates are selected from within the CEE in a process called 'Donor Discovery' (see *Figure 3*). Each person is checked regarding whether they have experience related to the URLs contained in the Contextual-Question. People that are found to be experiential-matches then form the counterpart to the Contextual-Question, which we call a Contextual-Answer. This answer not only contains knowledge about the URLs contained in the Contextual-Question, but also about that person's experientially-related URLs that are checked to be appropriately contextually-related to the URLs listed in the Contextual-Question.

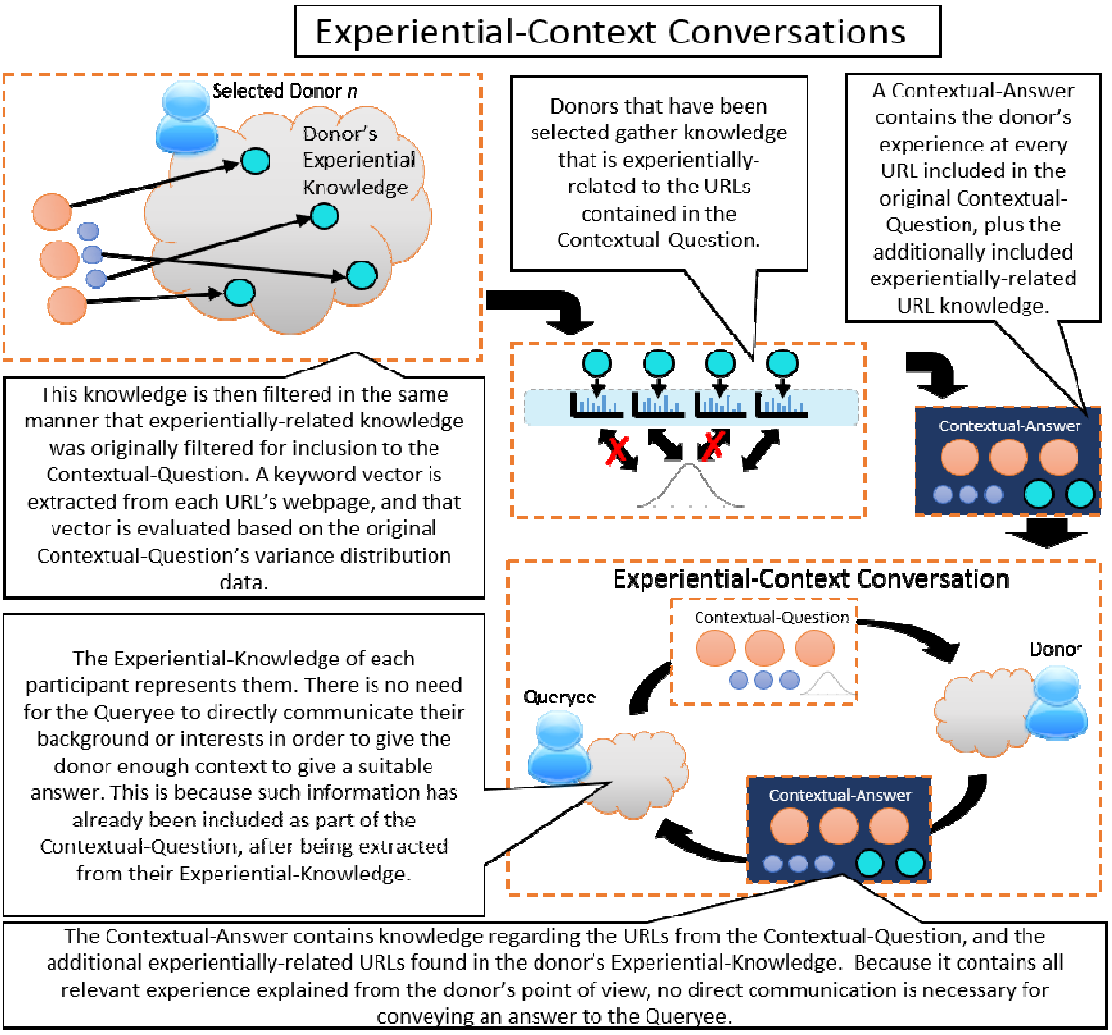

Figure 4. This figure describes the concept of Experiential-Context Conversations or ECConversations, which allow a context-based conversation between two people to take place using data-representations of those people in place of actual explicit conversation. Similar to how a real conversation would take place, the background experience and knowledge of the relative-novice are included in the question, and then the answer is tailored to the questions context, while still providing additional experientially-related knowledge from the experience-donor.

By encoding and adjusting for experience in each query, and for each potential match, a two-way Experiential-Context Conversation is taking place, similar to how a conversation between two people speaking to each other occurs. This is the reasoning behind the new keyword 'ECConversation' (see *Figure 4*).

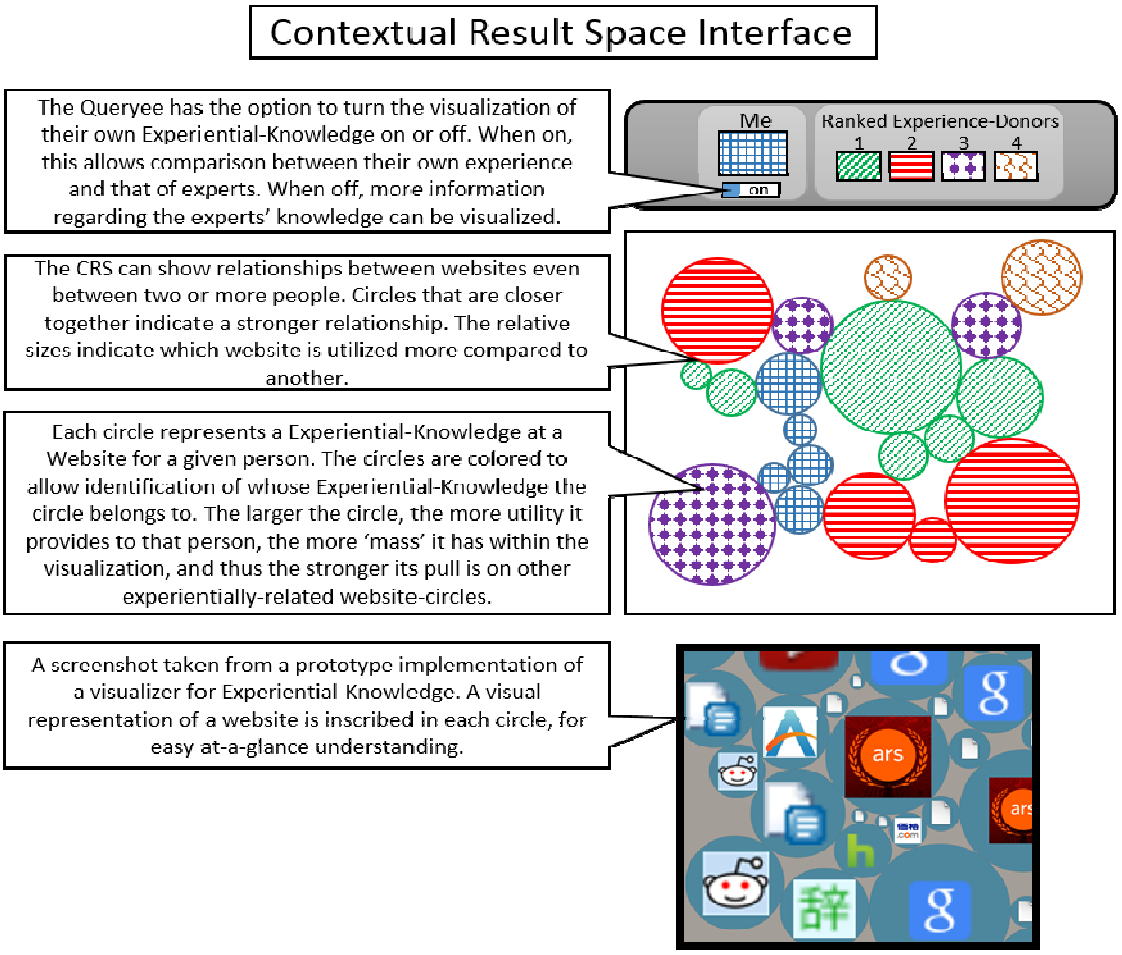

Figure 5. This figure describes the Contextual Result Space's (CRS) interface, whose main purpose is the visualization of the Contextual-Question and Contextual-Answers. The CRS uses circles to represent websites, and assigns them size, mass, and attractive forces to convey their relationships, and their relative-utility from their owners' point of view. Each person whose knowledge is represented in the CRS is assigned a different color, and thus the CRS provides a means to not only understand the value of knowledge-sources from others' points of view, but also to compare and contrasts those points of view.

As part of finding the appropriate people from which to retrieve Experiential-Knowledge as query results, relative-experience is calculated. This calculation is performed between a Contextual-Question, and a Contextual-Answer, which means only experience related to the relevant knowledge-domain is considered. In our model, usage of a URL equates to experience at that URL. Only people whose experience falls within a calculated range have their Contextual-Answers included in the results. Finally, a ranking function which calculates how close the ideal mix of experience at the desired information-sources, and relative-experience is performed, and the top few Contextual-Answers are returned. The need for a relative-expertise calculation is based on two premises:

- Experiential-Knowledge from people with similar or less expertise than the person performing the query is less likely provide usable and trustworthy insight when compared to a person with relative-expertise.

- Experiential-Knowledge from people with too much expertise may contain difficult-to-understand or difficult-to-utilize insights, due to the large gap in experience.

The results are returned in the form of a custom visualization we've created to express Experiential-Knowledge, called the Contextual Result Space (CRS). The CRS is an interactive visualization, which can express many simultaneous points of view, and allows for intuitive understanding of how those points of view relate amongst each other, including that of the person who performed the query. The goal of the results, and the subsequent visualization isn't to make a ranked list of 'best' URLs for the user, but to impart knowledge of how others use and value URLs on the Web, so that the person performing the query can make an informed decision on the value of that knowledge for themselves.

3.1. The Contextual Result Space

To express the complexity of an individual's Experiential-Knowledge, we have developed a custom visualization based on the RDCS [10] we call a Contextual-Result Space or CRS. The CRS represents Experiential-Knowledge of a website as a circle, adjusts the size (and consequently mass) of the circle based on the utility of the website, and models links in the Experiential-Knowledge graph as an attractive force between circles (see *Figure 5*). The final visualization-result is a result of a dynamic simulation of the interactions of the circles-representing-websites with each other.

The visualization receives Experiential-Knowledge as results from several donors, and combines it into one visual representation which allows intuitive comparison of knowledge between the Queryee's Contextual-Question, and the resulting Contextual-Answers. The CRS extends the functionality of the RDCS to better fit the realities of the Collaborative Experience Engine.
The CRS creates a space into which the Contextual-Question and top-ranked Contextual-Answers are combined. The basic rules which differ from the RDCSS are as follows:

- Each circle represents the Experiential-Knowledge of a person at a given URL, with respect to the other URLs present within the CRS. This means that any compiled experience which references a website that isn't also being represented in the CRS isn't counted. In this way, the utility of a website is properly conveyed within the context of the CRS being visualized.

- A given website can only be represented by one person's experience. This to keep the visualization clear and intuitive. If data from two or more people include experience about the same website, then the CRS chooses only one based on the ranking of the results, with the Queryee's Contextual-Question data being ranked the highest, and then the donors' Contextual-Answer data following in ranked order. This gives the most experientially-related knowledge the highest priority for knowledge-expression within the visualization.

- By default, the visualization of the Queryee's Contextual-Question data is also included, and takes the highest priority. This is so that the other Experiential-Knowledge can be understood in terms of it relating to the Queryee's experience. However, due to the priority rule, the visualization won't show any of the experts' experience for the websites that are already known to the Queryee. For this reason, there is a toggle that tells the CRS to remove the Contextual-Question experience from the visualization, so that the relative-experts' experience can be more completely visualized.

The visualization allows the Queryee to understand the importance and usability of knowledge-sources related to their query, from the point of view of relative experts.

4. CEE USAGE SCOPE

The Collective Experience Engine is a tool for learning about new or unfamiliar topics, and for when one must rely on new or unfamiliar data-sources to learn about those topics. It can be used in conjunction with any other tool or resource which provides information-sources-as-results. The most common of these is probably keyword-based search engines, however social networks, blogs, and expert-knowledge websites are all good candidates for being used in conjunction with the CEE, as they often contain references to additional potentially-useful knowledge-sources.

The CEE relies on the ability to capture a large portion of a person's experience in order to be effective. The Web is a good candidate for the CEE because all experiences between a person and the Web can be captured in that person's Web browser. Also, the list of all possible experiences dealing with Websites is limited to a few actions in the browser, such as loading a website, switching tabs, moving the mouse, scrolling, or spending time reading Web content.

5. ALGORITHMS AND THE DATA MODEL

This section describes the basic data structures and algorithms used within the CEE. A weighted-directed graph is used as a model for a person's Experiential-Knowledge. The primary algorithms include:

- A Contextual-Question formulator, which takes a query as input, and generates a Contextual-Question vector on-the-fly for searching against other peoples' Experiential-Knowledge.

- A Relative-Experience extraction and scaling algorithm. This algorithm allows for experience with respect to a given Knowledge-Domain to be calculated, and utilized as a query parameter. This works by allowing Contextual-Question vectors to be searched effectively, despite the original data having differing values due to relative differences in experience.

- The Contextual Result Space visualization algorithm, which allows for the comparison and contrast of a number of people's relevant Experiential-Knowledge simultaneously. This algorithm regulates the conversion of graph-data into placement, behavior, and visual aspects within the visualization that is utilized as the query results.

5.1. Modeling Experiential-Knowledge

We use an Experiential-Knowledge weighted-directed graph ($graph_{EK}$) to model per-user Web browsing activity within the CEE. Nodes in the graph represent a URL, often a Website (To denote a specific person's $graph_{EK}$, we utilize the notation $graph_{EK\text{-}id_{user}}$). The following information is represented in the graph:

5.1.2. URL Experience

All URLs experienced by a person are represented by the nodes within a person's $graph_{EK}$. The labeling function for the nodes in the graph uses a id_{url} and an id_{user} as inputs $node_{label} = f_{label}(id_{url}, id_{user})$. This means that while each user has their own independent nodes for the same URL, and thus conceptually each user has an independent $graph_{EK}$, all users' nodes have a globally unique identifier, and can be easily stored within a single namespace. Despite having globally unique labels, users' graphs can be easily merged for visualization purposes due to the use of common id_{url} ids.

5.1.3. URL View Transition

When a person changes the URL they are viewing--for instance when switching browser tabs, or opening a new window--a directed link is used to represent this activity. For instance, viewing URL_a, then URL_b results in the link $link_{ab}$ being created. Thus, *link.a* is the referrer or origin of a transition, and *link.b* is the target or endpoint of an activity.

5.1.4. URL Knowledge

A person's Knowledge of a URL is stored in the adjacency graph of $node_{label}$, which we call adj_{label}. The directed links stored in adj_{label} all include the time they were created (*link.then*).

5.1.5. URL Utility

A simple explanation of the Utility of a $node_{label}$ ($utility_{label}$) is that it is represented by its degree ($degree_{label}$) multiplied by the combined weight of the node's adjacency graph adj_{label}.

$$utilSimple_{label} = (label) \rightarrow deg_{label} * \left(\sum_{a \in adj_{label}} a.weight \right) \quad (1)$$

As a practical matter, the URL Utility isn't such a straightforward calculation. Firstly, the weight of a link isn't a simple property, but rather an aggregation function which we must calculate. We must also factor in both the time-decay of the links in adj_{label}, and the Knowledge-Domain that is being used as a context with which to understand the Experiential-Knowledge at a URL. A key for a link in adj_{label} is an *{a,b}* pair, and a *then* property, which defines when the link was created in the $graph_{EK}$. When factoring in time-decay, we use the following process to determine a decay-factor:

$$f_{decayFactor}(NOW, then) = \frac{1}{1 + \log\left(\frac{NOW - then}{1000}\right)} \quad (2)$$

Equation 2 describes a function which will take a *then* in UTC milliseconds, and return a weighting factor that will have no effect if *then* is the current time *NOW*, and will drop to approximately 0.067 in a year's time. We divide by 1000 in order to convert the units to seconds, and thus make the final units more consistent on a human scale. This equation favors recent knowledge more strongly than past knowledge. In the future, giving the user the ability to adjust the time-decay factor may be beneficial.

The utility calculation is affected by a selected Knowledge-Domain because the purpose of the Knowledge-Domain is to only allow Experiential-Knowledge relevant to the context of the query to be represented. A node in the $graph_{EK}$ might have many links in its adjacency graph that are irrelevant to the query at hand. To factor in the Knowledge-Domain, we simply perform an intersection between the set of id_{url} ids in the Knowledge-Domain, and the set of $link_{ab} \in adj_{label}$, using *b* to represent each link as we see in *Equation 3:*

$$validLinks = \{x \in adj_{label} \mid a.b \in knowledgeDomain\} \quad (3)$$

The *validLinks* set contains $link_{ab}$ links with a *link.a* and a *link.b* property where *b* is the target, and *a* is the referrer of a relationship between two $node_{label}$ nodes.

The final $util_{label}$ value with time-decay and Knowledge-Domain filtering is shown in *Equation 4:*

$$util_{label} = (label, NOW) \rightarrow deg_{label}\left(\sum_{a \in validLinks} f_{decayFactor}(NOW, a.then)\right) \quad (4)$$

5.2. ECConversation Algorithms

This section describes the algorithms necessary for the ECConversation functionality of the CEE. This includes how to formulate a Contextual-Question, using the Contextual-Question to find experience-donors for having an ECConversation with, and then how those donors generate a Contextual-Answer as a response.

The ECConversation Algorithms use two main factors in generating the data structure for querying: keyword relatedness, and utility-spread. The algorithm ensures that items extracted from a person's $graph_{EK}$ are related in terms of content to the explicitly selected query URLs' (id_{query}) ids by comparing the important keywords from those URLs' content. Later, Contextual-Question data structures (vec_{CQ}) are compared using fuzzy matching, by understanding the variance of utilities within the selected URLs, and using that variance as a range on a per- id_{url} id basis.

When a person submits a query, the system utilizes that person's $graph_{EK}$ to generate a Contextual-Question. A Contextual-Question is the data-structure used to ask the question 'What do other people know about information-sources regarding type-X, which are related to my previous experience?', and is used compare and search Experiential-Knowledge within the CEE.

5.2.1. Contextual-Question Formulation

The first thing that happens after a person submits a query, is the potential surrounding id_{url} ids are gathered ($id_{potentialSur}$). The $id_{potentialSur}$ ids are selected by being experientially-related to the id_{query} ids, within the $graph_{EK}$:

$$id_{potentialSur} = \left\{ \begin{matrix} x \in graph_{EK} | \\ x \notin id_{query} \wedge \{\exists link.b \in adj_{label} | link.b \in id_{query}\} \end{matrix} \right\} \quad 5$$

In order to understand whether a member of $id_{potentialSur}$ is contextually-related enough to the id_{query} id set, we must first calculate the mean and σ of the id_{query} URLs' keyword vectors. We use important keywords extracted from the source URLs represented by the id_{url} ids stored in id_{query} to do this. There are several potential algorithms with which to extract and calculate important keywords from a URL. We chose to use a modified form of tf-idf (term frequency-inverse document frequency). The function to extract a vector of scored keywords from a give URL with respect to a document set $document_{set}$ is $f_{keyscore}(id_{url}, document_{set})$. The $document_{set}$ is a set of ids, either $id_{potentialSur}$, id_{query}, or a combination of the two. This function returns a vector whose keys are keywords, and whose values are the scores of those keywords.

First we create a $documentSM_{vec}$ vector, containing the mean of all the vectors in the id_{query} ids $document_{set}$.

$$documentSM_{vec} = \sum_{a \in document_{set}} f_{keyscore}(a, document_{set}) \Big/ |document_{set}| \quad 6$$

Next, the mean and σ distance for individual $f_{keyscore}$ vectors from the id_{query} and the mean $documentSM_{vec}$ vector. The function f_{dist} is a standard distance formula calculation for measuring the distance between two vectors:

$$mean_{query} = \frac{\sum_{a \in id_{query}} f_{dist}(a, documentSM_{vec})}{|id_{query}|} \quad 7$$

$$\sigma_{query} = \sqrt{\frac{\sum_{a \in id_{query}} (f_{dist}(a, documentSM_{vec}) - mean_{query})^2}{|id_{query}|}} \quad 8$$

Now that both the $mean_{query}$ and σ_{query} distance values are calculated, we can select members from $id_{potentialSur}$ that are within ±1.5 σ_{query} from $mean_{query}$, and combine the resulting id_{sur} set with id_{query} to get our vec_{CQ}:

$$id_{sur} = \left\{ \begin{matrix} a \in id_{potentialSur} | \\ f_{dist}(a, documentSM_{vec}) > (mean_{query} - (1.5 * \sigma_{query})) \\ \wedge f_{dist}(a, documentSM_{vec}) < (mean_{query} + (1.5 * \sigma_{query})) \end{matrix} \right\} \quad 9$$

Now that we know which id_{url} ids the Contextual-Question is composed of, we can calculate the utilization statistics, so that we can do fuzzy matching when querying other people's Experiential-Data. We calculate the mean Contextual-Question id_{url} utilization $mean_{CQUtil}$ and its standard deviation (σ_{CQUtil}):

$$vec_{CQ} = id_{sur} \cup id_{query} \quad 10$$

Now that we know which id_{url} ids the Contextual-Question is composed of, we can calculate the utilization statistics, so that we can do fuzzy matching when querying other people's Experiential-

Data. We calculate the mean Contextual-Question id_{url} utilization $mean_{CQUtil}$ and its standard deviation (σ_{CQUtil}):

$$mean_{CQUtil} = \frac{\sum_{a \in vec_{CQ}} util_{flabel}(a, id_{user})}{|vec_{CQ}|} \quad 11$$

$$\sigma_{CQUtil} = \sqrt{\frac{\sum_{a \in vec_{CQ}} \left(util_{flabel}(a, id_{user}) - mean_{CQUtil}\right)^2}{|vec_{CQ}|}} \quad 12$$

The final Contextual-Question contains vec_{CQ}, $mean_{CQUtil}$, and σ_{CQUtil}.

5.2.2. Experience-Donor Search

Once we've generated the Contextual-Question, we use it to discover experiential-donors. As there is no keyword-based data remaining in the Contextual-Question, this search is done completely based on Experiential-Knowledge of URLs. The initial operation is simple--we form a subset of vec_{CQ}, based on subtracting 1.5 sigmas of σ_{CQUtil} from each key's value in vec_{CQ}. This removes any id_{url} ids that aren't absolutely required in order to describe the Contextual-Question, due to having a relatively-low utilization, and creates a base-requirement set (vec_{CQBase}) for matching the person's query:

$$vec_{CQBase} = \left\{a \in vec_{CQ} \mid \left(util_{flabel(a,id_{user})} - \left(1.5 * \sigma_{CQUtil}\right)\right) > 0\right\} \quad 13$$

Using the vec_{CQBase} set, we can then select the set of people within the CEE who are potential donor-candidates for providing Experiential-Knowledge to the person performing the query. Assume the CEE ($vec_{collective}$) is a set of $graph_{EK-id_{user}}$. In order to find the set of initial donor-candidates (vec_{init}):

$$vec_{init} = \left\{ \begin{matrix} a \in vec_{collective} \mid \\ (a.id_{user} \neq id_{user}) \\ \wedge (vec_{CQBase} \subseteq a) \end{matrix} \right\} \quad 14$$

5.2.4. Ranking and Result Limits and the Relative-Experience Algorithm

With too many results, the visualization would quickly become overwhelming. This means that the ability to rank, and then only select a limited number of experiential-donors' contributions to the results is important. The ranking algorithm sorts based on utilization-knowledge to create a vec_{rank} containing id_{user} keys and a utilization-score value where the greater the utilization-score, the greater the rank. Remember that $vec_{CQMatches}$ contains $graph_{EK}$ graph members:

$$vec_{rank} = (\forall a \in vec_{init}) \rightarrow \sum_{x \in (a.vec_{CQ} \cap vec_{CQ})} util_{flabel(x,a.id_{user})} - util_{flabel(x,id_{user})} \quad 15$$

In the case of performing a relative-experience calculation, we want the relative-ideal utility value to be based on a utility value an order of magnitude greater than the user, rather than the user's raw utility value. Thus, Equation 16 must be altered if such a calculation is desired:

$$vec_{rank} = (\forall a \in vec_{init}) \rightarrow \sum_{x \in (a.vec_{CQ} \cap vec_{CQ})} util_{f_{label}(x,a.id_{user})} - util_{f_{label}(x,id_{user})}{}^{10} \qquad 16$$

5.2.5. Generating Results

The purpose of the results is to describe to a person how other people utilize information-sources within the Contextual-Query's context. This means that we wish to return not just the matches to the Contextual-Question, but also the surrounding Experiential-Knowledge from the experience-donors' $graph_{EK}$. To do this, we perform identical operations to when the Contextual-Question was originally generated. The main differences are that instead of the id_{query} ids set, we instead substitute in the vec_{CQ} set, and of course the id_{user} user id is that of the donor's $graph_{EK}$. The calculations can stop after *Equation 10* is complete, and the resulting set is called $id_{QResult}$ instead of vec_{CQ}. These $id_{QResult}$ sets of id_{url} ids are then returned to the person who performed the query, and visualized.

6. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Our software architecture is client-server. The client is a Chrome browser plug-in, which is able to capture user-actions as they browse the web. These actions are fed to a server which compiles them into an Experiential-Knowledge graph. The data structures are represented in a NoSQL key-value based server called Redis. Communications are handled via a synchronization-based protocol provided by Firebase. The visualization is provided by a custom canvas-based HTML5 web application.

There are two main roles of the system:

- Passively mining web-behavior data, and converting it into the Experiential-Knowledge graph.
- Providing query functionality.

6.1. Web-Behavior Mining

While there are many web-behavior possibilities to capture and encode as Experiential-Knowledge data, our system focuses on loading URLs, and switching between tabs in the web browser. As the user performs these actions, the client plug-in captures them, and sends a message to the server with two pieces of information: the target URL, and the referrer URL (if there was one). For instance, switching between tabs causes the URL which was loaded in the previously-active tab to be the referrer, and the newly-activated tab's URL to be the target. The server then adds the Experiential-Knowledge to the person's graph.

As mentioned before, the Redis database uses a key-value system to store data, and provides data types such as strings, lists and sets. A node in the Collective-Brain is stored as two sorted-sets (for storing the links in the graph) and a per- id_{url} id sorted-set for storing the list of keywords and their scores for each URL represented in a person's $graph_{EK}$. For the sorted-sets representing the links, the key for a sorted-set is a concatenation of id_{user},id_{url}*[in\|out]*. In the implementation, links

are broken into incoming and outgoing lists, hence the "in" or "out" at the end of each node. The score of each entry in the sorted-sets is the time, and the value is an id_{url} id. Because there is a full set of incoming and outgoing data for each id_{url}, there are no dependency-lookups or chains, and thus data-retrieval is quick.

As an optimization, to allow for quicker matching amongst people, we also have a set that stores all of the id_{url} ids that are represented within a person's $graph_{EK}$. This allows us to perform a test to quickly determine if a given user has the entire set of core-URLs from the Contextual-Question, to qualify them as a potential-donor of experiential-knowledge for a given query.

6.2. Query Functionality

The Query interface and visualization is provided by a browser plug-in. Hooks in the browser's API allow us to modify the GUI to provide right-click menu options for adding URLs to a query. A button in the browser's chrome provides a person with the ability to see all of the currently-added URLs in a visual manner, and then submit the query. Results are visualized within the browser itself using a canvas-based HTML5 render, the same as the RDCS [10].

7. FUTURE WORK AND CONCLUSION

In the future, we plan on capturing more behavioral aspects of a user, including the amount of time spent viewing content, and capturing content embedded in a webpage. By saving each element embedded within a webpage individually, we can more accurately capture the experiential knowledge of a person, because we can discover relationships wherein the same content is embedded in multiple different webpages. Finally, the ability parse media in addition to text would further improve the accuracy of the CEE.

Another future improvement is the addition of query modifiers. For instance, allowing the Queryee to specify from what type of people (friends, experts, groups, a specific person) to query information. Perhaps a group would like to curate their own Experiential-Knowledge, or the Queryee would just like to ask friends since those people are considered more trustworthy. Another possible query modifier is the range of acceptable relative-experience.

Finally, a continuously working version of the CEE, where the Contextual Result Space updates as a user browses might provide a better experience, as the feedback would be quicker, and more tied to the browsing experience.

We have designed a system called the Collective Experience Engine which captures and distributes the experience of participants in the CEE for the collective benefit of the whole. The CEE overcomes the problem of insufficient knowledge about a topic causing uncertainty and poor choice when selecting knowledge-sources to learn from. The query process is designed to be easy and intuitive, automatically tailoring the query to the knowledge and experiential background of the person who submitted it. The use of Experiential-Context Conversations ensures that the content of the results are properly tailored to the context of the person submitting the query, and the relative-experience calculation ensure that the results are properly tuned to their experience level. The CEE targets a definite need on the Web for better dissemination of understanding of various knowledge-sources, and the ability to understand such sources from the point of view of people with more experience than oneself.

REFERENCES

[1] Susan R. Goldman, "Choosing and using multiple information sources: Some new findings and emergent issues ," Learning and Instruction , vol. 21, no. 2, pp. 238-242, 2011, Special Section I: Solving information-based problems: Evaluating sources and information, Special Section II: Stretching the limits in help-seeking research: Theoretical, methodological, and technological advances. [Online]. http://www.sciencedirect.com/science/article/pii/S0959475210000204

[2]Saskia Brand-Gruwel and Marc Stadtler, "Solving information-based problems: Evaluating sources and information ," Learning and Instruction , vol. 21, no. 2, pp. 175-179, 2011, Special Section I: Solving information-based problems: Evaluating sources and information, Special Section II: Stretching the limits in help-seeking research: Theoretical, methodological, and technological advances. [Online]. http://www.sciencedirect.com/science/article/pii/S0959475210000228

[3] Saskia Brand-Gruwel, Iwan Wopereis, and Yvonne Vermetten, "Information problem solving by experts and novices: analysis of a complex cognitive skill," Computers in Human Behavior, vol. 21, no. 3, pp. 487-508, 2005.

[4] Peter Gerjets, Yvonne Kammerer, and Benita Werner, "Measuring spontaneous and instructed evaluation processes during Web search: Integrating concurrent thinking-aloud protocols and eye-tracking data ," Learning and Instruction , vol. 21, no. 2, pp. 220-231, 2011, Special Section I: Solving information-based problems: Evaluating sources and information, Special Section II: Stretching the limits in help-seeking research: Theoretical, methodological, and technological advances. [Online]. http://www.sciencedirect.com/science/article/pii/S0959475210000198

[5] Charles A Perfetti, Jean-François Rouet, and M Anne Britt, "Toward a theory of documents representation," The construction of mental representations during reading, pp. 99-122, 1999.

[6] Suresh K and Jacob, Renju T and Nardine, Jennifer and Peck, Bhavnani, "Exploring the distribution of online healthcare information," in CHI'03 Extended Abstracts on Human Factors in Computing Systems.: ACM, 2003, pp. 816-817.

[7] M. Anne Britt and Cindy Aglinskas, "Improving Students'Ability to Identify and Use Source Information," COGNITION AND INSTRUCTION, vol. 20, pp. 458-522.

[8] Marc Stadtler and Rainer Bromme, "Dealing with multiple documents on the WWW: The role of metacognition in the formation of documents models," International Journal of Computer-Supported Collaborative Learning, vol. 2, no. 2-3, pp. 191-210, 2007.

[9] Ivar Bråten and Helge I. Strømsø and Ladislao Ladislao, "Trust and mistrust when students read multiple information sources," Learning and Instruction, vol. 21, no. 2, pp. 180-192, 2011.

[10] Jeremy Hall and Yasushi Kiyoki, "Creating a Personal-Context Oriented Real-Time Dynamic and Collaborative Space," Information Modeling and Knowledge Bases Xxiv, p. 82, 2013.