CollecTV Intelligence: a 3-screen 'social search' system for TV and video queries

Nitya Narasimhan, Myungcheol Doo, Joe Wodka and Venu Vasudevan

Abstract-Television and video consumption are growing rapidly worldwide, driving usage of second and third screens for related interactions and information. Since television has a proven impact on consumer purchase behaviors, there is commercial interest in technologies that model viewers' intent, interests and engagement. However, the ecosystem currently faces two challenges – audience fragmentation and a lack of integrated search features. Both have an impact on content providers' ability to track and profile their consumers and are thus critical to personalizing and monetizing content-related services. In this paper, we introduce CollecTV, a social search solution for TV and video queries that can mitigate these concerns by incentivizing the *consumer* to adopt consistent identities across three screens while also providing finer-grained attention and engagement analytics to content *providers*. We give an overview of the architecture and describe the evolution of the system from a TV-centric solution to a 3-screen model with native mobile and browser clients to maximize its usage and utility. We conclude by reviewing relevant research and outlining challenges and opportunities for further exploration in this space.

Index Terms—interactive television, collaborative question-answering, recommender systems, social search

I. MOTIVATION

TELEVISION and video consumption continue to increase steadily worldwide. Recent Nielsen studies [1] show that the average American consumes nearly 160 hours of television each month. Television is known to have sizeable influence on consumer awareness of products in the marketplace. Analysts looking at the impact of different media on consumer behavior across the purchase funnel noted that the television made a higher contribution to purchase behaviors than all other forms of media combined [2]. Unsurprisingly, this motivates industry interest in technology that models viewers' interests and intent. However, the TV ecosystem faces two key challenges today – *audience fragmentation* as viewers adopt more device-shifting and time-shifting models and a *search deficiency* that triggers multi-tasking behaviors and leads to attention fragmentation.

Audience Fragmentation: The Nielsen data also indicates that users consume content across domains – logging 36 hours on TV, 20 minutes online and 4 minutes on mobile (for video) and nearly 4 hours of Internet usage every week. Statistics differ by demographic, with older adults dominating television audiences and teenagers driving mobile video usage. Time-shifted viewing also increased, adding to the fragmentation of audiences for live TV. This has motivated providers to relinquish one-size-fits-all models for content scheduling and advertising in favor of targeted models focused on individuals. To deploy such models, providers need to track the consumers consistently across devices using some universal identity.

Search Deficiency: Despite pre-dating mobile and web use, TV lags its counterparts in providing users with search tools to find *information* related to onscreen content or other artifacts. Instead, most TV platforms support search UI only for content discovery (e.g., search by title). As a result, most users multi-task, using a second device concurrently to conduct queries or transactions, causing new problems for providers. Not only do they lose valuable analytics (from user queries) but they also lose revenue from related advertising and user purchases. They also have to deal with attention fragmentation as users switch their focus constantly between television and device screens. This motivates the need for an *integrated* search mechanism for TV that enables users to make queries or search for related information from within the viewing experience.

Three-Screen Usage: Though we emphasized television, we know that mobile and web domains play an increasingly important role in both content distribution and consumption. With more providers making their content available to mobile and web clients, we predict that consumers will soon demand such search and query capabilities uniformly across all three screens. We also see the lines blurred between the roles of the television 'as-device' and 'as-content'. Over-the-top models enable TV *content* to be delivered to Internet-enabled devices while Video-on-Demand solutions convert the TV into a rich-display *device* for new sources of video, beyond broadcast TV. Consequently, any search solution that we develop must also be extensible to support 3-screen usage.

This also has benefits. While shared TV viewing and big-screen displays promise lean-back experiences, resource-rich mobiles and PCs naturally favor lean-forward, interactive user behaviors. This can lead to two types of experience for search. In the *vertical* experience, a mobile or PC client can optimize search techniques for content in its domain; thus, a browser client can exploit tools like 'query suggestions' to reduce user effort. In a *companion* experience, a mobile or PC client can act as a second screen to TV, supplementing the lean-back

N. Narasimhan, J. Wodka and V. Vasudevan are in the Applied Research Center of Motorola Mobility, Inc. They can be contacted by email at {nitya, joe.wodka, venu.vasudevan}@motorola.com.

M. Doo is currently a PhD student at Georgia Institute of Technology. He can be contacted at mcdoo@cc.gatech.edu.

viewing experience with a personalized, contextually-coupled "interaction" interface. With our CollecTV[1] system we aim to support both models. Thus, the core goals of CollecTV are (a) to offer a more intuitive search experience for TV and video content, (b) to extend the solution to mobile and web clients, enabling both vertical and companion experiences on 3 screens and (c) to incentivize users to adopt a consistent identity across the three screens for analytics and personalization services.

The rest of this paper describes the design and development of the CollecTV system, starting from our decision to adopt a *social search* strategy for user-initiated queries, to prototyping various client experiences (TV, web and mobile) that explore new value propositions for search in rich media environments.

II. ADVOCATING 'SOCIAL SEARCH'

Users watching television or video content invariably have questions about featured artifacts (things, people). Because TV platforms today lack an integrated search capability, users are forced to conduct these queries online – either on search engines or within collaborative question-answering (CQA) communities like Yahoo! Answers. The advantage of this is it allows us to analyze the archived Q&A or search engine query logs to gain a better understanding of what users ask and why.

We find questions that are objective (*'Who is X?', 'What is X' or 'Where is X?'*), but also others that are subjective (*'Who does X remind you of?'*). Questions also reveal user intent, be it simple information (*'How much is it?'*) or a precursor to a transaction (*'Where can I buy it?'*). We present a sample query in Figure 1 to illustrate two problems users face in pursuing a split-device strategy for TV search. The first is query creation (what to ask), the second is query routing (where to ask it).

Figure 1: A sample question from Yahoo! Answers.

Query creation. In many cases, the subject of the query is a specific onscreen artifact – e.g., the 'shirt' in the example. But describing visual context is hard. Doing so in 4 words or fewer (the optimal query length for search engines) is impossible. As a result, we find users struggling to articulate their query to optimize their chances of finding the right answer. This user took the added effort to locate an online screenshot featuring the shirt of interest. However, not every user will have the time to search for images, or the good fortune to find the right one.

Query routing. A user's first instinct is to query a search engine but these require good keyword selection and involve user effort in scanning pages of results for useful answers. A visual search engine may be useful if users possess an image of the artifact. However, such engines are in their infancy and not

always accurate or reliable. And users may not always have an image for comparison. Search engines will excel at *non-visual* queries, particularly if they are objective (e.g., ask about show schedules, characters or plot-points). They may succeed on some subjective queries if sufficient user-generated content exists that can be surfaced as possible opinions on that topic. For instance, Google query suggestions for the TV show 'Lost' place *'What does it mean?'* near the top; query results mostly link to blogs and tweets speculating on just this topic.

For *visual* queries however, we need a different approach. We make two observations. First, a picture is worth a thousand words – i.e., query creation is simple if the user could only attach a screenshot of that artifact to his query for context. And second, humans have the edge over machines when it comes to image recognition [3].We not only overlook rendering defects but we can extrapolate intent by intuitively filling n the gaps on missing context or information. We also have a domain advantage in TV-related information given inherent interest in television trivia, entertainment news and celebrity culture.

These insights led to our decision to adopt a social search model in CollecTV. In this context, social search refers to the crowd-sourcing of queries to users or communities, relying on the wisdom of crowds to generate at least one useful answer. While this originated with CQA portals like Yahoo! Answers and Stack Overflow, it has since evolved to more *interactive* 'engines' like Vark [4] and services like ChaCha [5]. Their key difference is strategy. CQA forums favor a destination model where the *users* find queries to answer. Interactive approaches favor a distribution model where the *query* finds its way to the user. In our system, we build on CQA research [6, 7] but adopt the query distribution model. Our decision was influenced by two factors: (a) rapid adoption of social networking services provides increased ways to identify relevant 'experts' to target, and (b) a 3-screen solution affords more ways to present that query to the targeted user.

III. SYSTEM ARCHITECTURE

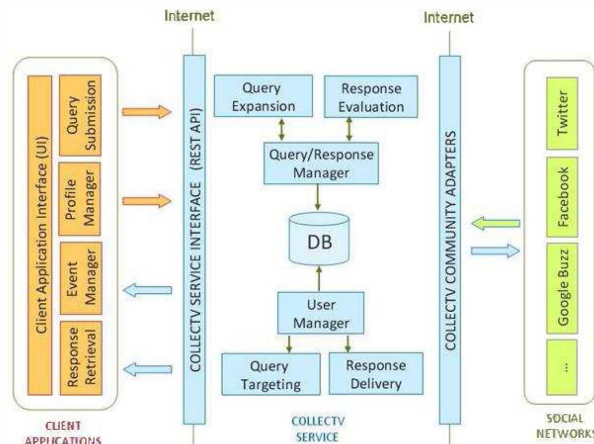The CollecTV system employs a client-server architecture as shown in the figure below.

Figure 2: The CollecTV System Architecture

---

[1] Pronounced 'collective' to indicate crowd-sourced search for TV

In this section, we outline briefly, different components of this architecture and the role they play in enabling a social search experience. Clients reside on user-facing devices like the TV (set-top box), mobile and PC and provide the user interface to the service. The server resides in the cloud and is essentially the 'engine' performing the social search. In this context, it employs two approaches. First, it interfaces to third party communities and social networks like Twitter, enabling it to *propagate* queries to targeted users within them. Second, it uses the clients to *recommend* suitable queries to its own user population based on profile and context matching.

The Client: The primary role of the client is to assist users in creating queries, and in retrieving or viewing responses. In that context, it leverages domain- and device- specific features to reduce user effort in performing these tasks. In addition, the client also performs background tasks that assist the server in making query recommendations to this user. This involves two responsibilities. First, it updates the server interactively with information related to the user's current viewing context; this helps build the user profile. Second, it retrieves suitable query recommendations to present to the user in the current context.

On devices that support push notification capability, clients may also incorporate an event manager to process real-time notifications from the server (instead of proactive fetch or poll behaviors). This improves the user experience by (a) enabling immediate notifications of responses, and (b) making the query recommendation process more efficient.

The Server: The server is the crux of our system providing the social search 'engine' for television and video queries. It does four things: interaction management, data management, and query routing and response curation. The first relates to communication with clients and third-parties (social networks, online forum, even search engines) to fulfill user queries. The second refers to managing access to stored information including user profile, and queries and their related responses. However, the real 'brains' of the operation are in query routing and response curation – both fertile areas for further research. From a design standpoint, these elements are represented by the Query Targeting and Response Evaluation modules. We designed these to support pluggable algorithms, allowing us to try out different strategies and mechanisms for recommending queries or rating (and filtering) responses. We will show a few examples of these later, when describing client prototypes.

In addition, we also have two helper modules for Query Expansion and Response Delivery. The first associates a user query with a unique identifier that can be used by recipients to access additional context; this helps us meet payload limits set in third-party APIs used to route queries to users. We also see potential for use in providing query suggestions to users during query creation on the client. The Response Delivery module has the simplest task – to determine when, where and how to return responses to the user. Criteria may include destination choices, delivery strategies or privacy preferences.

We walk through representative CollecTV experiences in the next few sections, highlighting client implementations on

TV (set-top box), PC (browser) and mobile (Android) in order. We have deliberately focused on describing only a subset of features on each platform, favoring breadth over depth in order to showcase the broader value proposition of this system. *Caveat*: Client screenshots are from early research prototypes; we are currently redesigning both interfaces and experience to deliver a consistent look-and-feel across three screens, and to incorporate new capabilities and some experimental features.

IV. *TV ANSWERS*: A SET-TOP BOX EXPERIENCE

Unsurprisingly, our first client was targeted squarely at the TV platform, particularly at the "Set-top box" that typically delivers content and experiences for consumption via an attached television display. We called that experience *'TV Answers'* to highlight its roots in collaborative Q&A solutions. A first version of that system is described in more detail in our earlier paper [8]. In this paper, we will simply highlight some key features; the figure below shows a version of our query and response UI for this experience on the set-top box.
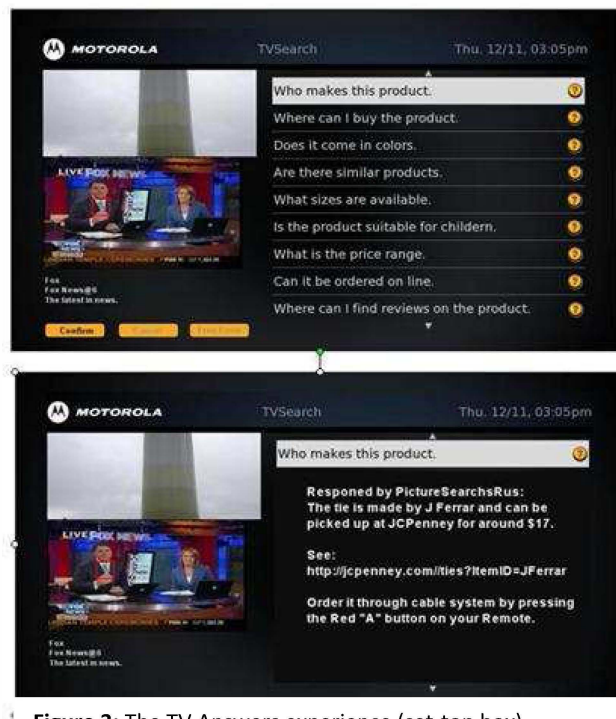

Figure 3: The TV Answers experience (set-top box)

As we mentioned earlier, the primary duty of the client is to reduce user effort in making the query. This includes entering the query text (the actual question) and any supporting context that would help route or clarify that query. On the TV/STB platform, text entry often involves a remote control and virtual keyboards – not conducive to enabling descriptive queries. On the plus side, the set-top box has rich program context (meta-data like title, genre and cast) that can substantially help both in reducing manual data entry (and error) during creation and in context-matching for routing. In addition, we also simplify query creation in two innovative ways as shown in Fig. 3 (a).

The first is *Freeze-frame*, a utility we provide on the set-top

box that allows users to simply pause the screen and highlight the area of interest with a provided 'cursor'; the utility stores this marked-up frame on the server, and annotates the query with the related reference (URL). Seekers and recipients can now see a more complete picture of the query and its context.

The second is *Query-templates*, a query suggestion utility we provide that presents users with the most probable queries for the given context. Selecting from a list is easier to do with a remote than explicit text entry. And, standardized queries can be used to build a knowledge base around that topic that may directly answer the user's request. Thus, query creation is now effectively a 3-step procedure: freeze-and-select the query subject (image), select the query text (template), and submit. Users can always opt to generate a query without visual annotations and using manual text entry.

Because of current platform limitations, most set-top boxes cannot support push notifications; consequently, we adopted a 'refresh on-demand' approach where seekers can pull up the interface seen in Figure 3 (b) to check for responses. This can cause delays in user viewing of responses and raise privacy concerns in shared devices like TV. In the mobile companion experience (described later) we provide one solution to these issues. However, we note that it is possible to support basic privacy enablers (e.g., PIN to unlock) directly on TV.

Finally, this lack of inherent push capability on TV also hampers our ability to recommend queries to TV viewers in an effective manner; while we can use a pull strategy, it requires users to initiate the refresh without having any incentive to do so. We are exploring other ways to incentivize the user and tackle this issue. However, in our earlier prototype, we chose to route queries primarily to online communities and social networks like Twitter for user responses.

V. *TV SEARCH*: A BROWSER EXPERIENCE

Our next prototype focused on the PC, targeting the web browser in particular as a client platform. We saw two primary opportunities here – support search around *web video* content and target Internet users for query *responses* by recommending them queries that are contextually relevant. The first illustrates a *vertical* experience for the domain (web) while the second hints at a *companion* experience for other domains like TV. The browser plug-in approach interested us for three reasons – first, over-the-top strategies and Internet TV portals like Hulu ensure that television content is increasingly available online. Second, browsers are invariably indicative of individual (vs. shared) viewing, thus enabling targeted experiences. Third, the browser is by nature an interactive medium, making it more likely for users to engage in both query creation and response activities while consuming content.

Figure 4 shows screenshots of our first plug-in experience, developed using the add-on capability of the Firefox browser. We note that such 'extensions' are now the norm for standard browsers including Microsoft's Internet Explorer and Google's Chrome. We highlight the following features in this prototype before entering into an interesting discussion on challenges.

First is *query creation*. For the vertical experience, unlike the television (where we need to physically store the frame), the web is already hyperlinked and accessible. Thus, a user watching online video needs simply to provide the URL for the content as relevant context. But how about narrowing down the 'frame' of reference? Unfortunately, online portals often employ different container strategies for delivering a video-player experience – and not all of them make the player events observable by our plug-in. Thus, we can tell how long the user stayed on a site and how many times he visited it – but cannot always narrow down the specific frame or content offset at which a query was made. Fortunately, the HTML5 standard offers promise of more transparency, and trends indicate that it may see widespread adoption amongst online content portals.
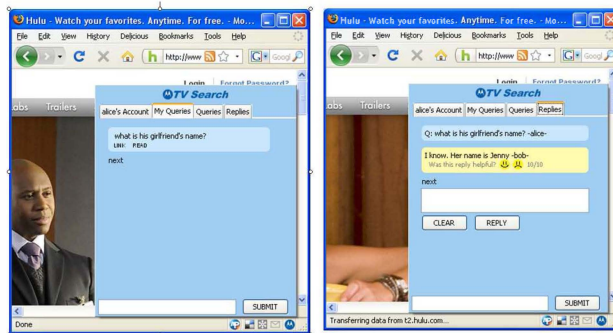

Figure 4: The TV Search plug-in UI for query and response

Another aspect of query creation is non-visual context capture. On the television, this takes the form of metadata provided by an electronic program guide (EPG) or similar service. On the web, every content portal defines metadata in its own way, or in some cases, not at all. A survey of the top 10 Internet TV and online video sites showed that the lowest common denominator was program title and publication date, though sites like Hulu and YouTube provide richer metadata like descriptions and categories. Such inconsistencies made it difficult not just to annotate the query (for efficient routing) but also to recommend queries (for response) to that user. We explored ideas around meta-data *harmonization* and extrapolation that we defer to a future discussion.
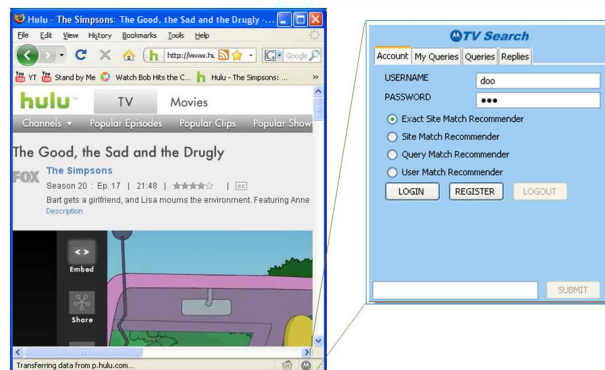

Figure 5: The TVSearch plug-in: user login UI

Another issue is *query recommendation*. Because the user is

online and always-accessible, it becomes easy for us to exploit technologies like Ajax to enable real-time content updates to the interface. In our case, this involves two steps – first, the plug-in provides current context (user profile, content URL) to the server which uses the information to add that user (and client) to its list of query targets. Next, if the user clicks on the 'Queries' tab, he instantly sees query recommendations based on his current context. Switching now to a different video will simply cause the screen to refresh with new (more relevant) queries based on the choice or recommender algorithm. As seen in Figure 5, we are also exploring different strategies for query recommendation such as *exact match* (originated on the same webpage), *site match* (originated on the same web portal, but on a different page), *query match* (is for similar content but from a different portal) and *user match* (is from a user with similar tastes in content).

Note that query recommendation necessitates tracking the user's activities on the web, raising a genuine privacy concern. We address this with an ambient indicator (the Motorola icon at bottom right of the browser) which is activated when users log into the client. The typically-gray icon turns blue whenever tracking data is being uploaded to servers. Users log out of the plug-in to prevent tracking. Further, we *only* report data if the user is on a known content portal site – thus, tracking may be active on Hulu but disabled on Google News or CNN sites.
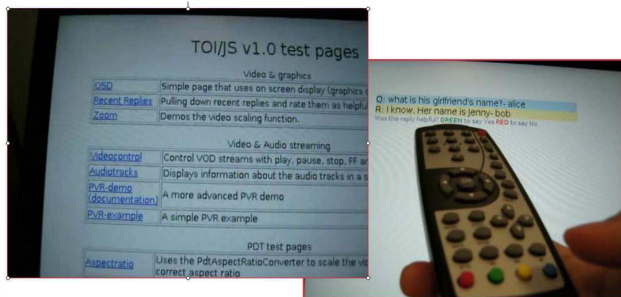

Figure 6: The 'Recent Replies' evaluation app on TV

Finally, we touch briefly on *Response Evaluation*. A critical downside of social search is that the answer quality is tied closely to answerer knowledge and interest. How can we know that a response is correct? As shown in Figure 5, we allow the seeker to rate various responses to reflect his *satisfaction* with that reply. However, the seeker may not know if the answer is in fact correct. Moreover, when we have queries that elicit multiple answers, we often need to rate or rank them in order to deliver utility without cognitive overload to users. For the web client, we explored an interesting concept in cross-domain validation. Basically if a web query relates to content that is also available on TV, can we now tap into the TV audience to validate the response – effectively crowd-sourcing the evaluation step. We note that while TV viewers are poor choices for *answering* questions (due to input constraints), they are ideal for *voting* on responses. Figure 6 shows a simple demo of this concept, where we deployed an IPTV application that fetched TV-related query-answer pairs from the server and

presented them for a vote to the user. Users pressed a button on their remote to register their vote (Yes or No) at the server. Responses with substantially more 'yes' votes are prioritized for user review over unrated or marginally-correct responses. Note that this is an area requiring more research to be truly effective since web and TV domains use different metadata formats and values. With richer correlation or harmonization mechanisms, we can also improve the evaluation process by targeting TV viewers.

VI. *TV COMPANION*: A MOBILE EXPERIENCE

Finally, we present our mobile client prototype. While we used the web prototype to explain support for domain-specific 'vertical' search extensions, we use the mobile prototype to describe the 'companion' experience. We emphasize again that we are discussing the relevant *subset* of capabilities for each client and not the comprehensive solution.
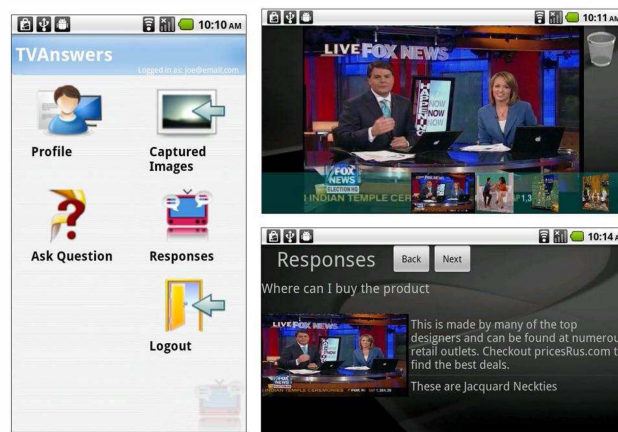

Figure 7: The TV Answers mobile 'companion' experience

The mobile client provides two value propositions for users of the CollecTV system. First, it is a personal device and is therefore an ideal destination for delivering responses in real time and without the usual privacy implications. Next, it acts as a secondary *interaction* screen while users consume content on a primary display device like TV or PC. Figure 8 shows a very preliminary prototype for this experience that illustrates some of these capabilities. The key challenge here is in context sharing between the primary and secondary screens in order to ensure an effortless and intuitive experience to the user.

First, we look at the basic requirements – query creation and response delivery. Because we designed the CollecTV service to be RESTful [9] and deployed it in the cloud, it is instantly accessible to any Internet-enabled devices. Mobile devices, in particular smart-phones and tablets, fall into this category. Thus, creating and posting simple text queries is trivial. However, to support visual queries, or to enable automated annotation of content meta-data, we need communication between the primary and secondary screens. This has both a research and a business challenge, and is under exploration. But, we can allow users to retrieve previously-created images from the server and edit or create a query against those visuals.

As shown in Figure 8 (top right image), the 'Images' tab of the mobile client presents a carousel of thumbnails to the user for selection; images are a combination of self-generated, third-party generated and provider-supported thumbnails. Because these images come from legitimate TV sources, they have associated program context that serves as useful metadata for query annotations. We predict that more intuitive context-sharing approaches will emerge thanks in no small part to the increased usage and popularity of *Social TV* applications (e.g., GoMiso, Tunerfish or HotPotato) that allow users to engage with other users virtually (online) around currently watched programs or content.

VII. RELATED WORK

In the previous sections we motivated the need for a social search engine (and service) for supporting user-defined queries around TV and video. And we describe our architecture and aspects of client platforms to support a three-screen experience around this concept. This has provided the basic *plumbing* for our solution and validated our ability to address basic needs in query creation and response retrieval. However, the real value and differentiation of such services lies in their *routing* and *grading* intelligence. This is the focus of our ongoing research and we build on a large body of work from the web domain.

In this context, research in collaborative question-answering communities is of immediate interest. As the volume of queries increases, query prioritization is critical, involving a trade-off between starvation and utility. Issues like *seeker satisfaction* [6] and *time-sensitivity* [7] are invaluable for such decisions. From the response perspective, understanding the criteria that drive quality answers [10] or reliable feedback ratings [11] are useful for efficient and accurate evaluation of responses.

We are also greatly influenced by both research and services in the *social search* space. Services like Vark [4] and ChaCha [5] have brought crowd-sourcing into the mainstream, enabling users to both *submit* queries *and* receive responses through third-party services like Twitter. They differ in their use of routing intelligence. Vark monitors social graphs and chatter across many networks to create expertise profiles that enable more precise query targeting; their use of social graphs as an incentive to users to respond is also novel. ChaCha however routes queries to a team of paid 'experts' with diverse domain expertise. These services are complementary, not competitive, to our CollecTV system; routing a subset of queries to such services simply increases the probability of getting a response.

Finally, we also benefit from work in the human factors space that looks at the *motivation* of users [10] in turning to social networks instead of search engines. Interestingly, such insights also work well from understanding the *recipient*'s point of view – in other words, what incentivizes a user to take the effort to answer others questions. Such ideas complement more game-theoretical approaches to designing incentives for participatory behavior [13].

VIII. CONCLUSIONS AND NEXT STEPS

Increased growth in video and television consumption along with x-shifting behaviors is causing audience fragmentation and motivating the need for mechanisms that track consumers across three screens. At the same time, consumers lack critical search capabilities using free-form queries in the TV and video domains. We believe that delivering a compelling 3-screen personalized search strategy is vital to incentivizing consumers to adopt a consistent identity across the three screens; it also facilitates finer-grained attention and engagement analytics for users through analysis of their free-form queries.

This drove our design and implementation of the CollecTV system, a three-screen strategy that uses the collective wisdom of crowds to resolve user queries with rich media contexts. We presented early client prototypes of our work to illustrate the CollecTV value in serving both vertical domain requirements, and horizontal companion experiences. Our next challenge is to facilitate context-sharing and metadata harmonization for enabling cross-domain query recommendations. We will also explore both machine-learning and human factors approaches to motivating active user participation within such systems.

REFERENCES

[1] "What Consumers Watch: Nielsen's Q1 2010 Three Screen Report, *http://blog.nielsen.com/nielsenwire/category/reports-downloads/*

[2] "Study shows TV Impact on Consumer Purchasing Behavior", Apr 2009 *http://www.tvweek.com/news/2009/04/study_shows_tvs_impact_on_cons.php*

[3] L. von Ahn and L. Dabbish, "Labeling Images with a Computer Game", Proceedings of Conference on Human Factors in Computing Systems (SIGCHI), New York, 2004, pp. 319–326

[4] D. Horowitz, S. P. Kamvar, "The anatomy of a large-scale social search engine", Proceedings of the 29th International Conference on the World Wide Web (WWW), Raleigh NC, 2010, pp. 431-440

[5] ChaCha, *http://chacha.com*

[6] Y. Liu, J. Bian and E. Agichtein, "Predicting information seeker satisfaction in community question answering", Proceedings of the 31st International ACM SIGIR Conference, Singapore 2008, pp 483-490

[7] Y. Liu, N. Narasimhan, V. Vasudevan and E. Agichtein, "Is this urgent? Exploring time-sensitive information needs in collaborative question-answering", Proceedings of the 32nd International ACM SIGIR Conference, Boston MA, 2009, pp 712-713

[8] N. Narasimhan, J. Wodka, P. Wong and V. Vasudevan, "TV Answers: using the wisdom of crowds to facilitate searches with rich media context", Proceedings of the 7th IEEE Conference on Consumer Communications and Networking Las Vegas NV, pp 749-753

[9] R. Fielding, "Architectural Styles and the design of Network-based Architectures", PhD Thesis, University of California at Irvine, 2000.

[10] M. R. Morris, J. Teevan and K. Panovich, "What do people ask their social networks, and why?: a survey study of status message q&a behavior", in Proceedings of the 28th International Conf. on Human Factors in Computing Systems (CHI), Atlanta, GA, 2010, pp 1739-48.

[11] F. M. Harper, D. Raban, S. Rafaeli, J. A. Konstan," Predictors of answer quality in online Q&A sites", Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, Florence Italy, pp 865-874.

[12] J. Bian, Y. Liu, E. Agichtein and H. Zha, "A few bad votes too many?: towards robust ranking in social media", Proceedings of the 4th international workshop on Adversarial information retrieval on the web, Beijing China, 2008, pp 53-60.

[13] S. Jain, Y. Chen and D. C. Parkes, "Designing incentives for online question-answer forums", Proceedings of the tenth ACM conference on Electronic commerce, Stanford CA, 2009, pp. 129-138.