A Theoretical Framework for Information Search using Online Social Networks

Rohit Negi
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
negi@ece.cmu.edu

Mustafa Yilmaz
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
mustafay@andrew.cmu.edu

***Abstract*—A significant part of human activity today consists of searching for pieces of information online, an endeavor that may be time-consuming if the individual searching for the information is unfamiliar with the subject matter. However, online social networks exist where experts can aid individuals by answering questions posed by these users. This paper describes a theoretical framework to model the dynamic process by which requests for information arrive in a social network of experts, who then answer the requests after a random time interval. Results on the requests-handling capacity of the network are provided.**

***Index Terms*—information search, scheduling, social network, community question answering.**

I. Introduction

Social networks are an integral part of modern life, connecting friends and family, and also serving as fast sources of breaking news. Researchers are exploring theoretical underpinnings of online social networking platforms in terms of activity engagement, topic interest and user experience. Applications of crowdsourcing in online platforms has recently become more prominent and attracted researchers to study mechanisms to incentivize workers to complete tasks [1]. Information exchange and diffusion on online social networks have far-reaching societal implications and researchers have begun studying processes through which information diffusion takes place, its implication for opinion formation and measurement [2], methods for maximizing diffusion [3], and mathematical models are being developed to understand its effects on collective human behavior and psychology [4].

In this paper, a different type of application of social networks is explored. Individuals today must deal with a number of small problems, which require each individual to search the Internet for ideas relevant to solving each problem - an activity that may receive mixed results depending on the expertise of that individual. But given pervasive online connectivity, there are potentially a large number of 'experts' available online that an individual can consult, who can contribute their knowledge to solving problems related to their expertise. Community Question Answering (CQA) forums [5] such as `Quora` and `StackExchange` [6] , and on a smaller scale, `Piazza` used for teaching courses, connect users to experts, and are real-life examples of such a system. Questions posted on these social networks are answered by experts according to their fields of expertise and interests.

Previous work on question routing in social networks focuses on the algorithmic aspects of identifying experts who can answer questions in the shortest amount of time based on expertise and possible response time [7]. However, there is currently no work that models question-answering in networks abstractly with a view to analyze their capacity to handle requests. By assigning specific questions, which are simply *requests* for information, to individual experts or groups of collaborating experts, questions can be answered more rapidly and effectively, potentially increasing their capacity. To this end, this paper envisions a large number of requests for information being made to a given social network, but also a large number of potential experts available to answer those requests. Since the requests must be responded to in a timely manner, we propose a dynamic framework, where requests arrive stochastically, are handled by expert(s) who search for relevant information utilizing their own knowledge, and depart when the expert provides a response. In this initial work, results on scheduling requests, and on the resulting capacity for handling requests are presented for a fully connected social network such as `Quora`, with arbitrary networks considered in future work.

II. Theoretical Framework

The problem setting in this paper assumes that requests for information come into a social network stochastically. Each request (question), is handled by an expert, who searches for information to answer that request, augmented with her own knowledge, and succeeds in providing information answering that request after a random amount of time, based on the complexity of the request. This requires describing a quantitative model for information search (or complexity of the request) and also describing a model for scheduling these requests, so that experts can answer them.

1) **Model of Information Search**
Time is assumed to be discretized finely, so that it is measured as $t = 1, 2, 3, \ldots$ time slots. Let $\mathcal{M}$ be a large set of information facts. A *topic* $x \subset \mathcal{M}$ is a large

This material is based upon work supported by the US National Science Foundation under Grants 1422193 and 2008570.

subset of facts, e.g., 'Windows 10 debugging'. The set of topics $\mathcal{X}$ may be large but is assumed to be finite to avoid technical clutter. An *expert* is a *research time* function $T : \mathcal{X} \rightarrow [1, \infty)$, where $T(x)$ is the mean time that the expert takes to answer a request concerning topic x; this average time is assumed to be known to the expert, although the expert does not apriori know the random time it will need to answer a specific request. This time is required because the expert will typically need to search for information relevant to the specific request and/or think deeply about the request using her knowledge before being able to answer it. We assume the number of slots to answer a specific request is a geometrically distributed random variable with values $1, 2, \ldots$ (and mean value $T(x) \geq 1$). A typical request may be 'Why does my Windows 10 laptop become hot and shut down?', which concerns the topic 'Windows 10 debugging'. For conciseness, we sometimes call a request concerning topic x as *request* x, if the context is clear.

2) **Model of Dynamic Scheduling**

It is assumed that there is a social network of n experts labeled $1, 2, \ldots, n$, represented as a directed graph $G = (V, E)$, where the vertices V represent experts and the directed edges E represent coordination opportunity between pairs of experts. By *coordination*, we mean that a scheduler (described below) can assign a request in expert i's queue to expert j, as long as $(i, j) \in E$ in the graph. For example, if experts exclusively use a CQA forum such as `Quora`, they can all coordinate with each other, i.e. G is a complete graph (each pair of vertices has edges in both directions.) On the other hand, if a social network like `Twitter` or `Facebook` is used, the graph may have a complex structure, precluding arbitrary coordination. This paper only considers a complete graph linking the experts, so as to elucidate the key ideas of the framework, with arbitrary graph connectivity considered in a future paper. See Figure 1.

We adopt a dynamic stochastic model of information searching. At the end of each time slot t, requests in different topics $x \in \mathcal{X}$ may arrive in the system, and so, we need a multi-class queuing model. Here, by system, we mean the complete graph social network that all the experts are monitoring for requests. Denote as $a_x(t) \geq 0$ the number of requests in topic x arriving at the end of time slot t, with the average number of requests arriving being $E[a_x(t)] = \lambda p(x)$. Here, $p(x) > 0$ is a probability mass function (p.m.f.) over topics x (so, $\sum_{x \in \mathcal{X}} p(x) = 1$), which causes requests for certain topics to appear more frequently. $\lambda \geq 0$ is the *request load* in the system, i.e., average number of requests arriving in each slot. We assume that the variance of $a_x(t)$ is upper bounded by $c_a \lambda^2$ for all x for some constant c_a. The arrival in topic x is independent of arrival of requests in other topics, and arrivals in other time slots.

A scheduler then assigns different requests waiting in the queues to different experts, subject to the social network graph, allowing the experts to *coordinate* in handling the requests. For the case of complete graph considered in this paper, the scheduler can assign any request to any expert. Expert i works on its assigned request x (equivalently, called 'researches x') by searching for information or thinking about the request utilizing her knowledge, and answers it successfully in that time slot with probability $q_i(x) \doteq \frac{1}{T_i(x)} \leq 1$. So, $q_i(x) > 0,\ \forall\, x, i$ is the *answering rate* at which expert i answers requests in topic x, and indicates the expertise of that expert for topic x. Experts with larger $q_i(x)$ presumably have deeper knowledge that allows them to quickly research problems, and so, a crude measure of expertise of an expert is the arithmetic mean answering rate $R_i \doteq \frac{1}{|\mathcal{X}|} \sum_x q_i(x)$. We assume that the scheduler knows the answering rates $q_i(x)$ for different topics, but it does not apriori know the random time needed by an expert to answer a specific request. $d_{x,i}(t) = 0, 1$ indicates failure or success of expert i in finding the answer for a request in topic x during time slot t, respectively. If the request is not answered, we assume that the expert continues work on the request until it is answered in some subsequent slot. Thus, the number of time slots needed by the expert to work on a request before successfully answering it is indeed a geometric random variable with average time $T_i(x)$. Since arrivals occur at the end of a time slot, the topic queue lengths update as $Q_x(t+1) = Q_x(t) + a_x(t) - \sum_i d_{x,i}(t), \forall x$.

We will consider *offline* schedulers, by which we mean schedulers that know the topic p.m.f. $p(x)$ of the arrival process, perhaps by estimating it using past arrival history. To show stability, we will only consider schedulers that use the current state, which is the set of current queue lengths $Q_x(t), \forall x$ before new requests arrive in that slot, to decide the schedule for time slot t. Due to this, and since the arrivals are independent in time and the research time is a geometric distributed (i.e., memoryless) random variable, the request answering system is a Markov chain. We say that the system is stable if the Markov chain of queues is positive recurrent [9]. To define instability, we allow any arbitrary scheduler that uses past history, not necessarily only current queue lengths. We call the system unstable if, for any choice of such a scheduler, the sum of the queue lengths diverges with non-zero probability, i.e.,

$$P(\lim_{t \to \infty} \sum_x Q_x(t) = \infty) > 0. \tag{1}$$

Note that the probability in (1) is zero for a positive recurrent Markov chain. A request load λ for which the system is stable under some scheduling policy, is called *achievable*. The supremum of achievable loads is called *capacity*.

All proofs are in the appendix. We will generally omit writing the set of an index, when that set is obvious, e.g., writing $\sum_x$ instead of $\sum_{x \in \mathcal{X}}$. Similarly, we will write $\max_{\alpha_i}$ instead of $\max_{\alpha_i, i=1,2,\ldots,n}$.

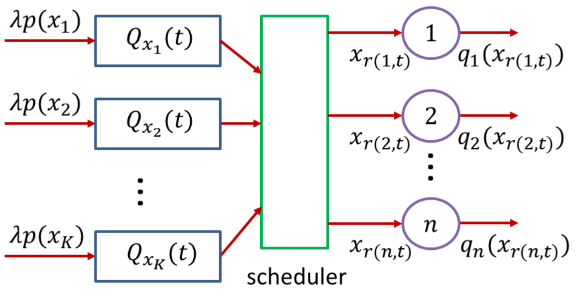

Fig. 1: Coordinating experts in complete graph.

III. Results

Based on the theoretical framework of information search presented in Section II, we present preliminary results on the performance of the system.

A. Homogeneous Experts

Consider a simple setting of n homogeneous experts. i.e., for each expert i, assume $q_i(x) = q(x)$, $\forall x \in \mathcal{X}$.

Lemma 1: The capacity with homogeneous experts in a complete graph is $\lambda^* = n\left(\sum_{x\in\mathcal{X}} \frac{p(x)}{q(x)}\right)^{-1}$. Further, any $\lambda < \lambda^*$ can be achieved using a scheduler that assigns any idle expert an arbitrary request from the queues. In particular, for $n = 1$, we get the capacity of a single expert.

From the lemma, capacity is high if the expertise of the homogeneous experts matches closely with the population of requests coming in, so that none of the ratios $\frac{p(x)}{q(x)}$ is too large. Specializing to $n = 1$, capacity λ^* in the lemma is simply the harmonic mean of the answering rates $q(x)$ of that single expert with respect to the arrival p.m.f. $p(x)$ (the standard harmonic mean essentially uses the uniform p.m.f.) In light of this elementary result, we can call $\lambda_i^*(p) \doteq \left(\sum_{x\in\mathcal{X}} \frac{p(x)}{q_i(x)}\right)^{-1}$ as the capacity of expert i with respect to p.m.f. $p(x)$, which is a refinement of the arithmetic mean R_i measure in Section II, but now accounting for the arrival statistics.

Given the large number of topics (large $\mathcal{X}$), we may be willing to reject requests that do not match the expertise available to research them. Let $e_x(t) \leq a_x(t)$ be the number of newly arriving requests on topic x that are rejected at time t. We wish to characterize homogeneous-experts capacity under $\varepsilon-$loss constraint, i.e., the maximum load that these experts can handle while keeping queues stable but with fractional loss bounded as below.

$$\limsup_{T\to\infty} \frac{\sum_{t=1}^{T}\sum_x e_x(t)}{\sum_{t=1}^{T}\sum_x a_x(t)} \leq \varepsilon \quad \text{almost surely.} \tag{2}$$

Let $|A|$ denote size of set A. For the next result, re-index the topics so that $q(x_1) \geq q(x_2) \geq \cdots \geq q(x_{|\mathcal{X}|})$. Then, calculate $N, \mu(x_N)$, with $1 \leq N \leq |\mathcal{X}|$, $0 < \mu(x_N) \leq 1$, as the unique solution to the following equation.

$$\mu(x_N)p(x_N) + \sum_{j=1}^{N-1} p(x_j) = 1 - \varepsilon. \tag{3}$$

Lemma 2: With the topics re-indexed as above, if we are willing to accept fractional loss $\varepsilon < 1$, the capacity of homogeneous experts in a complete graph is $\lambda^* = n\left(\mu(x_N)\frac{p(x_N)}{q(x_N)} + \sum_{j=1}^{N-1}\frac{p(x_j)}{q(x_j)}\right)^{-1}$, with the $N, \mu(x_N)$ calculated as the unique solution to (3). Further, any $\lambda < \lambda^*$ can be achieved by the scheduler shown below.

Offline Lossy scheduler: The offline scheduler in Lemma 2 first calculates $N, \mu(x_N)$ before considering requests. After that, as requests come in, in each slot t, the scheduler assigns any idle expert one request arbitrarily from among those waiting in the topic queues. Subsequently, for the $a_x(t)$ requests for each x that come in at the end of that slot, the scheduler inserts them into topic x queue if $x \in \{x_j, j \leq N-1\}$, inserts them into the topic x_N queue with probability $\mu(x_N)$ if $x = x_N$, and drops them all if $x \in \{x_j, j \geq N+1\}$.

For $\varepsilon = 0$, the capacity specified in Lemma 2 is the same as in Lemma 1, because the equality (3) can only be satisfied by $N = |\mathcal{X}|, \mu(x_N) = 1$ since $p(x) > 0$ is a p.m.f. Lemma 2 is especially useful when there is a gross mismatch between the requests and the homogeneous experts. For example, if $q(x) = 0$ iff $x \in \mathcal{X}_0$ for some set $\mathcal{X}_0$, the lossless capacity is $\lambda^* = 0$. But if we allow loss, we can get capacity of $n\left(\sum_{x\notin\mathcal{X}_0}\frac{p(x)}{q(x)}\right)^{-1} > 0$, while accepting a fractional loss of $\varepsilon = \sum_{x\in\mathcal{X}_0} p(x)$.

B. Heterogeneous Coordinating Experts

A mis-match between the topic p.m.f. and the answering rates of the homogeneous experts in Section III-A can significantly lower capacity. Recruiting heterogeneous experts, which have different topics of specialization (i.e., different functions $q_i(x)$), may allow them to compensate for each others' weaknesses, and so, the capacity in the heterogeneous coordinated experts case is of interest. But then, the scheduler will need to send an expert requests that lie in her topics of expertise, i.e., for which she has small average research time $T_i(x) = \frac{1}{q_i(x)}$.) However, these average research times may be erroneous, since they may have been obtained as estimates based on past answering history. So, it is also of interest to guarantee stability when scheduling with erroneous estimates $\hat{T}_i(x)$ of the true $T_i(x)$, under appropriate assumptions on the magnitude of these errors.

Lemma 3: The capacity with heterogeneous coordinating experts in a complete graph is at least equal to the following lower bound.

$$\lambda^* = \left(\max_{\alpha_i}\sum_{x\in\mathcal{X}}\min_i\left(\alpha_i\frac{p(x)}{q_i(x)}\right)\right)^{-1} \quad \text{where} \tag{4}$$

$$\sum_{i=1}^{n}\alpha_i = 1, \quad \alpha_i \geq 0,\ \forall i = 1,\ldots,n. \tag{5}$$

Further, any $\lambda < \lambda^*$ can be achieved using an offline scheduler, such as the one shown below. Also, for some constant $\gamma \leq 1$, if the offline scheduler below uses erroneous research times $\hat{T}_i(x) \geq \gamma T_i(x), \forall x, i$, then any $\lambda < \gamma\lambda^*$ can be achieved, where λ^* in (4) was also calculated using these $\hat{T}_i(x)$.

Offline Coordinating scheduler: The scheduler is assumed to know $p(x), q_i(x)$. It maintains separate topic queues $Q_{x,i}(t)$ for each expert i. Before considering requests, the scheduler first calculates the solution to the convex dual problem [10] of the maximization problem over α_i stated in (4). (we will refer to this maximization problem as problem (4).) The dual problem is the Linear program below (see Lemma 5).

$$\mu^* = \min_{\mu,\, s_{x,i}} \mu \quad \text{s.t.} \tag{6}$$

$$\sum_{x \in \mathcal{X}} \frac{p(x)}{q_i(x)} s_{x,i} \leq \mu, \ \forall i \tag{7}$$

$$\sum_i s_{x,i} = 1, \ \forall x, \quad s_{x,i} \geq 0, \ \forall x, i. \tag{8}$$

In each time slot, the scheduler assigns a request arbitrarily to expert i from among the requests queued up at that expert's queues $Q_{x,i}$. Expert i is kept idle if and only if her own queues are all empty. Then in the same slot, using the above precomputed $s_{x,i}$ (which we note is a p.m.f. over i for each x), for each arriving request x in that slot, the scheduler selects an expert i randomly and independently according to the p.m.f. $s_{x,i}$, and then appends that request into the topic queue $Q_{x,i}$ of the selected expert i.

As opposed to homogeneous expert scheduling, in this case, any one expert mismatched to the request p.m.f. $p(x)$ may not be catastrophic. In fact, the following case shows that a *diversity* of experts may be preferable. Suppose there are a large number n of experts, with each expert i having expertise $R_i = \frac{1}{|\mathcal{X}|}\sum_x q_i(x) = \frac{1}{|\mathcal{X}|}$. Consider a toy case where $|\mathcal{X}| = n$ topics labeled $x_1, x_2, \ldots, x_n$ and $p(x) = \frac{1}{n}, \forall x$. If the experts are homogeneous, i.e., $q_i(x) = q(x), \forall i$, then the capacity in Lemma 1 is $\lambda^* = n\left(\sum_{x\in\mathcal{X}} \frac{1}{nq(x)}\right)^{-1} \leq n\frac{1}{n}\sum_x q(x) = n\frac{1}{|\mathcal{X}|} = 1$, by using the fact that harmonic mean is no more than arithmetic mean. Instead, suppose we have diverse experts with $q_i(x) = 1-\varepsilon$ if $x = x_i$, else $\frac{\varepsilon}{n-1}$, where ε is a small positive constant, each of which also has expertise $R_i = \frac{1}{|\mathcal{X}|}$ as in the case of homogeneous experts. Then, the capacity lower bound using Lemma 3 (noting that the optimal $\alpha_i = \frac{1}{n}, \ \forall\, i$ here) is $\lambda^* = n(1-\varepsilon)$, which is more than the homogeneous experts case. The advantage of diversity holds in general, as shown below.

Lemma 4: Capacity lower bound λ^* of heterogeneous experts in Lemma 3 is bounded as follows.

(a) *Coordination benefit*: $\lambda^* \geq \sum_{i=1}^n \lambda_i^*$, where λ_i^* are the individual expert capacities (defined in Lemma 1 by setting $n = 1$) when they pick requests without coordinating with each other.

(b) *Diversity benefit*: $\lambda^* \geq \lambda_{hom}^*$, where λ_{hom}^* is the capacity of the homogeneous experts calculated in Lemma 1 for (homogeneous) experts having answering rate $q(x)$ that is the average answering rate of the (diverse) experts, i.e., $q(x) = \frac{1}{n}\sum_{i=1}^n q_i(x), \ \forall\, x \in \mathcal{X}$.

IV. Simulations

To demonstrate the utility of our theoretical analysis, we performed simulations using Python. We arrange a set of $|\mathcal{X}| = 125$ topics on a regular 3-dimensional grid of length 2 on each side, where closer topics are interpreted to be similar, and assumed a uniform $p(x)$ p.m.f. We assume $n = 125$ coordinating experts, with the answering rate $q_i(x)$ of each expert i following a truncated Multivariate Normal distribution, with mean values on a regular 3-dimensional grid of length 2 on each side, and the standard deviation β controlling the dispersion of her expertise among the topics, normalized so that each expertise $R_i = \frac{1}{|\mathcal{X}|}$, where R_i was defined in Section II. This model is similar to that assumed by methods that match a low dimensional vector representation of the text of a question to a similar representation of experts to assign the question to that expert [11]. Note that in this model the peak answering rate is higher when experts are specialized (small β) at the expense of narrower expertise over the topics. Figure 2 shows how the coordinating capacity bound λ^* in Lemma 3 changes when the expertise dispersion β changes for $n = 27, 64, 125$ experts. Note that the capacity peaks at a certain optimal β, for which the experts are neither too broad nor too specialized. For larger number of experts, the optimal β is smaller, since experts are available to cover topics even when they are specialized. In the same figure, we also show the capacity for $n = 125$ uncoordinated experts (discussed in Lemma 4(a)), as well as $n = 125$ homogeneous experts (each with answering rate specified in Lemma 4(b)). In both cases, the capacity is lower than the $n = 125$ heterogeneous coordinated experts case.

Next, we run queuing simulations for $|\mathcal{X}| = 125$ topics and $n = 125$ heterogeneous coordinated experts modeled as in the previous paragraph, but now with requests scheduled dynamically by the Offline Coordinating scheduler, to observe the effect of load λ on the queue lengths over a period of 10^5 slots. The topic queue length $\sum_{i=1}^n Q_{x,i}(t)$ is averaged over time and over 5 runs, and its maximum among all topics is plotted in Figure 3 for various loads and expertise dispersion values β. We observe that queue lengths are small as long as the request load λ is relatively low and increases the closer we get to the capacity bound λ^* specified in Lemma 3 (shown by vertical lines), with the queues becoming unstable at or beyond that bound. In a real system, pushing the request load beyond capacity will result in a large number of requests remaining unanswered, potentially causing users to abandon the social network.

V. Conclusions

This paper set up a theoretical framework to analyze the dynamic process by which requests for information arrive in a social network, so that a set of experts can answer those requests. Capacity results were obtained for offline schedulers that assume knowledge of request arrival p.m.f. $p(x)$, for the case of a complete graph social network such as a CQA forum. These results quantified the importance of matching expertise available to the request topics, and also of the need

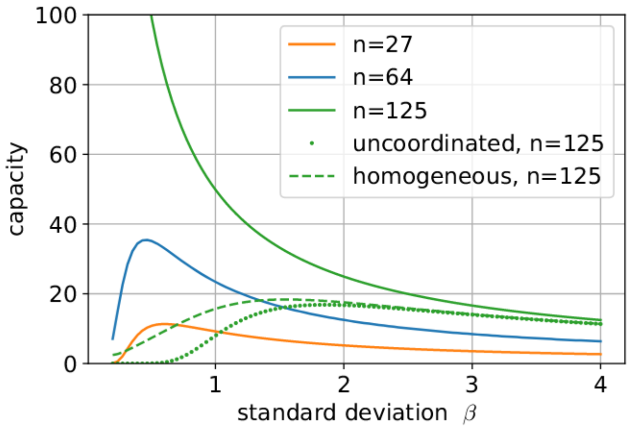

Fig. 2: Capacity versus standard deviation β.

for diversity of expertise. Future work will look at online schedulers that do not need to know $p(x)$, and will also social networks with arbitrary graph structure.

Appendix

Proof [Lemma 1]: This queuing system is essentially the discrete-time equivalent of an M/G/n queue with the service time (i.e., average answering time) being $\sum_x \frac{p(x)}{q(x)}$, from which the capacity result can be obtained. Alternatively, we can use Lyapunov analysis to show this. Let $\mathbf{Q}(t) \doteq [Q_x(t)]$ be the vector of topic queue lengths. $\mathbf{Q}(t)$ is an (irreducible) Markov chain because the scheduler only uses the current queue lengths for scheduling (see the argument in Section II). To show that any load $\lambda < \lambda^*$ is achievable using the specified scheduler, consider the Lyapunov function $L(t) = \sum_x \frac{1}{q(x)} Q_x(t)$ for $\mathbf{Q}(t)$. Then, $E[L(t+1) - L(t)|\mathbf{Q}(t)] = \lambda \sum_x \frac{p(x)}{q(x)} - \sum_i \sum_x \sigma_{x,i}(t)$, where $\sigma_{x,i}(t) = 1$ if the scheduler has assigned at time t a request from topic x to expert i, else 0. Let $F = \{\mathbf{Q}(t) : \sum_x Q_x(t) < n\}$. Since the specified scheduler is work conserving, $\sum_x \sigma_{x,i}(t) = 1, \ \forall i$ if $\mathbf{Q}(t) \notin F$ because each idle expert can be assigned a (different) queued request to work on. So, for the case $\mathbf{Q}(t) \notin F$, $E[L(t+1)|\mathbf{Q}(t)] = L(t) + \lambda \sum_x \frac{p(x)}{q(x)} - n = L(t) - \delta$, where $\delta \doteq n - \lambda \sum_x \frac{p(x)}{q(x)} > 0$ since $\lambda < \lambda^*$. For the case $\mathbf{Q}(t) \in F$, $E[L(t+1)|\mathbf{Q}(t)] \leq \frac{n}{\min_x q(x)} + \sum_x \frac{\lambda p(x)}{q(x)} < \infty$. Considering both cases, by Foster-Lyapunov theorem [9], the irreducible Markov chain $\mathbf{Q}(t)$ is positive recurrent, which proves stability.

For the converse, i.e., instability if $\lambda > \lambda^*$, we defer to the converse of Lemma 2, since this lemma is a special case of the former with $\varepsilon = 0$. □

Proof [Lemma 2]: As N increases, the second term in LHS of (3) is strictly monotonically increasing in $[0, 1)$ since $p(x) > 0, \forall x$ is a p.m.f. So, for given $0 < 1-\varepsilon \leq 1$, there is a unique $N \in \{1, 2, \dots, |\mathcal{X}|\}$ for which that term is (a) strictly less than $1 - \varepsilon$ while (b) it would equal or exceed $1 - \varepsilon$ for a larger N. Defining $\mu(x_N) = \frac{1}{p(x_N)}(1 - \varepsilon - \sum_{j=1}^{N-1} p(x_j))$, the two conditions imply that $0 < \mu(x_N) \leq 1$, showing that (3) indeed has this (unique) solution.

To analyze the Offline Lossy scheduler to prove achievability when $\lambda < \lambda^*$, note that as far as the queues are concerned, it simply modifies the arrival rates into the queues to $\lambda p(x_j)\mu(x_j), \forall j$ with $\mu(x_j) = 1, \mu(x_N)$,0, according to as j is less than, equal to, or more than N, respectively. So, by Lemma 1, the capacity is indeed at least the λ^* specified in this lemma, and is achieved by the specified lossy scheduler. For the losses, denote the decision to drop all the requests in topic x at time t by $v_x(t) = 1$, else 0. So $e_x(t) = v_x(t)a_x(t)$, with $E[v_x(t)] = 1 - \mu(x)$. Since $a_x(t), v_x(t)$ are i.i.d. and independent of each other, by Strong Law of Large Numbers (SLLN), almost surely (a.s.), the fractional loss $\limsup_{T\to\infty} \frac{\sum_{t=1}^T \sum_x e_x(t)}{\sum_{t=1}^T \sum_x a_x(t)} = \frac{\sum_x E[v_x(t)]E[a_x(t)]}{\sum_x E[a_x(t)]} = \varepsilon$ due to (3), as required by (2).

For the converse, suppose $\lambda > \lambda^*$, so that $\delta \doteq \lambda(\lambda^*)^{-1} - 1 > 0$. We allow any arbitrary scheduler that uses past history here, i.e., not necessarily only current queue lengths. The total amount of research time that must be expended by the experts for all requests that arrive within the first T time slots and are not rejected (which would create loss) is $W(T) = \sum_x \sum_{t=1}^T \sum_{k=1}^{a_x(t)-e_x(t)} W_x(t,k)$, where the i.i.d. variables $W_x(t,k)$ represent the time needed to research the k^{th} request of topic x arriving at time t. However, the total research time the n experts could have devoted within T slots is at most nT. So, due to SLLN, the amount of research $\tilde{W}(T)$ still waiting in the queues at time T a.s. satisfies $\liminf_{T\to\infty} \frac{1}{T}\tilde{W}(T) \geq \liminf_{T\to\infty} \frac{1}{T} W(T) - n = \liminf_{T\to\infty} \sum_x E[W_x(t,k)] \frac{1}{T} \sum_{t=1}^T (a_x(t) - e_x(t)) - n = \liminf_{T\to\infty} \sum_x \mu_x(T) \frac{\lambda p(x)}{q(x)} - n$, where $\mu_x(T) \doteq \frac{\sum_{t=1}^T (a_x(t) - e_x(t))}{\sum_{t=1}^T a_x(t)} \in [0, 1]$. Assume that the scheduler keeps the system stable, so that the probability in (1) is zero. Since the scheduler does not know the research time of a request when it schedules it, and since the research times are all finite a.s., the probability that $\tilde{W}(T) \to \infty$ is also zero. But from the above inequality, this can only occur if a.s. we have $\liminf_{T\to\infty} \sum_x \mu_x(T) \frac{\lambda p(x)}{q(x)} - n \leq 0$. i.e., there must be an infinite set $\mathcal{T} = \{T_1, T_2, \dots\}$ of time instants T for which

$$\sum_x \mu_x(T) \frac{\lambda p(x)}{q(x)} - n \leq 0. \tag{9}$$

Suppose for some $T \in \mathcal{T}$, we have $\sum_x \mu_x(T)p(x) \geq 1 - \varepsilon$. Choosing $\mu(x) \in [0, 1] \ \forall x$, with the constraint that $\sum_x \mu(x)p(x) \geq 1 - \varepsilon$, to minimize $\sum_x \mu(x) \frac{\lambda p(x)}{q(x)}$ results in optimal $\mu(x_j) = 1, 0$ according to as j is less than, or more than N, respectively, where $N, \mu(x_N)$ are the solution calculated from (3). Thus, for that T, we must have $\sum_x \mu_x(T) \frac{\lambda p(x)}{q(x)} - n \geq n \frac{\lambda}{\lambda^*} - n = n\delta > 0$, by using the definition of λ^*. Since this contradicts (9), there must exist some $\varepsilon' > \varepsilon$ (which depends on δ but not on T) such that $\sum_x \mu_x(T)p(x) \leq 1 - \varepsilon', \ \forall T \in \mathcal{T}$. So then, by S.L.L.N. $\limsup_{k\to\infty} \frac{1}{T_k} \sum_{t=1}^{T_k} \sum_x e_x(t) = \limsup_{k\to\infty} \sum_x (1 - \mu_x(T_k)) \frac{1}{T_k} \sum_{t=1}^{T_k} a_x(t) =$

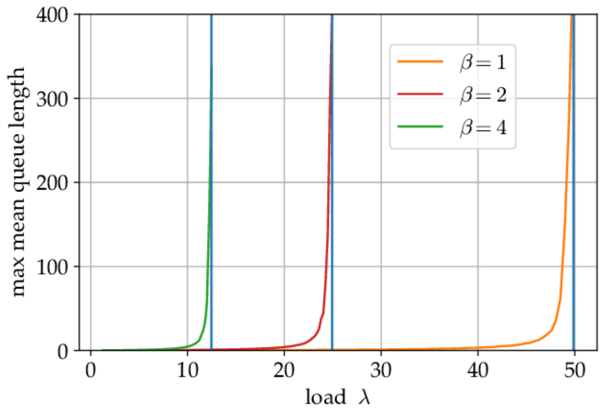

Fig. 3: Maximum average queue length at different loads.

$\limsup_{k\to\infty} \sum_x (1 - \mu_x(T_k))\lambda p(x) \geq \varepsilon' \lambda$. Thus, the fractional loss $\limsup_{T\to\infty} \frac{\sum_{t=1}^{T}\sum_x e_x(t)}{\sum_{t=1}^{T}\sum_x a_x(t)} \geq \varepsilon' > \varepsilon$, exceeds the allowed loss. This converse result, along with the achievability result above, shows that λ^* specified in this lemma is the lossy capacity. □

Lemma 5: The problem (6) is the convex dual of problem (4), and Strong duality applies. i.e., $\mu^* = (\lambda^*)^{-1} < \infty$.
Proof: The problem (4) can be written as

$$\max_{\alpha_i,\ \beta(x)} \sum_{x\in\mathcal{X}} \beta(x), \quad \text{s.t.} \tag{10}$$

$$\beta(x) \leq \alpha_i \frac{p(x)}{q_i(x)}, \ \forall x, i, \tag{11}$$

$$\sum_{i=1}^{n} \alpha_i = 1, \quad \alpha_i \geq 0.$$

With $s_{x,i} \geq 0$ being the dual variables for inequalities (11), the Lagrangian is $J = \sum_x \beta(x) - \sum_i \sum_x s_{x,i}(\beta(x) - \alpha_i \frac{p(x)}{q_i(x)})$. Maximizing the Lagrangian over $\beta(x)$ shows that (8) is obtained. Then, maximizing the Lagrangian over α_i shows that (6) and (7) are obtained, thus obtaining the dual problem.

Note that since $\alpha_i \leq 1, \forall i$, we get $\min_i \alpha_i \frac{p(x)}{q_i(x)} \leq \frac{p(x)}{\max_i q_i(x)}, \forall x$, so that the reward in the primal maximization (4) is no more than $\sum_x \frac{p(x)}{\max_i q_i(x)} \leq \frac{1}{\min_x \max_i q_i(x)} < \infty$, where finiteness is due to the assumption that $q_i(x) > 0, \forall x, i$. So, the primal maximization (4) has optimum value $(\lambda^*)^{-1} < \infty$, and so, the Linear program satisfies Strong duality. □

Proof [Lemma 3]: We prove achievability, i.e., stability of system if $\lambda < \lambda^*$, where λ^* is the solution to (4). Let $s_{x,i}, \mu^*$ be the optimal solution of the dual problem (6). Recollect that the offline scheduler uses this optimal $s_{x,i}$ to assign requests to experts' individual queues. For expert i, the arrival of $a_{x,i}(t)$ requests into its queue $Q_{x,i}$ is independent of arrivals of other topic requests to its own queues and to other experts' queues, and has a rate of $\lambda p(x) s_{x,i}$. So, the load at expert i is $\lambda_i = \sum_x \lambda p(x) s_{x,i}$ with topic p.m.f. $p_i(x) = p(x)s_{x,i}(\sum_x p(x)s_{x,i})^{-1}$. Since the scheduling of expert i only considers its own queues, its schedule is independent of schedules of other experts. So we can analyze the queue stability of each expert i separately. Setting $n = 1$ in Lemma 1, we know that expert i's capacity alone is $\lambda_i^* = \left(\sum_x \frac{p_i(x)}{q_i(x)}\right)^{-1} = (\sum_x p(x)s_{x,i}) \left(\sum_x \frac{p(x)s_{x,i}}{q_i(x)}\right)^{-1} \geq (\sum_x p(x)s_{x,i})\lambda^*$, where the last inequality is by (7) and Lemma 5. Since we also have $\lambda < \lambda^*$, we get $\lambda_i^* > \lambda_i$, and so, Lemma 1 (specialized to $n = 1$) guarantees stability for queues of expert i. Since this is true for each i, the system is stable, thus proving the first part (concerning using true $q_i(x)$).

For the second part of the lemma, call $\hat{q}_i(x) = \frac{1}{\hat{T}_i(x)}$. Suppose the offline scheduler uses the erroneous $\hat{T}_i(x)$ to calculate the fractions $s_{x,i}$ in problem (6) and also to calculate the capacity λ^* using (4). Assume that load $\lambda < \gamma\lambda^*$. Then, for those $s_{x,i}$, we get using (7) for all i that $1 \geq \lambda^* \sum_x \frac{p(x)}{\hat{q}_i(x)} s_{x,i} \geq \gamma\lambda^* \sum_x \frac{p(x)}{q_i(x)} s_{x,i} > \lambda \sum_x \frac{p(x)}{q_i(x)} s_{x,i}$. Here, the first inequality is by Lemma 5 because the primal optimal $(\lambda^*)^{-1}$ in (4) and the dual optimal μ^* in (6) were both computed using the same erroneous $\hat{q}_i(x)$. So, $1 > \lambda \max_i \left(\sum_x \frac{p(x)}{q_i(x)} s_{x,i}\right) = \lambda(\lambda^*)^{-1}$. The last equality identifies the true capacity λ^* of the system using Lemma 5 and the first part of Lemma 3 (which uses the true $q_i(x)$). Thus, $\lambda < \lambda^*$ where capacity is for the true $q_i(x)$, and so, by the first part of this lemma, the system is stable. □

Proof [Lemma 4]: Part (a) is intuitively obvious, so we skip the formal proof, which involves using $\lambda_j^* \left(\sum_x \min_i \left(\alpha_i \frac{p(x)}{q_i(x)}\right)\right) \leq \alpha_j \lambda_j^* \sum_x \frac{p(x)}{q_j(x)}$, and also showing that $\sum_{i=1}^{n} \lambda_i^*$ is the system capacity without coordination. For part (b), since $\min_i \frac{a_i}{b_i} \leq \frac{\sum_i a_i}{\sum_i b_i}$ for non-negative numbers a_i, b_i, we get from (4) that $(\lambda^*)^{-1} \leq \sum_x p(x) \frac{\sum_i \alpha_i}{\sum_i q_i(x)} = \frac{1}{n} \sum_x \frac{p(x)}{q(x)} = (\lambda_{hom}^*)^{-1}$, where the first equality uses (5) and the definition of $q(x)$. □

REFERENCES

[1] C. Huang, H. Yu, J. Huang and R. Berry, "Crowdsourcing with Heterogeneous Workers in Social Networks," *IEEE Globecom*, pp. 1-6, Dec. 2019.

[2] T. Tong and R. Negi, "Supermajority sentiment detection with external influence in large social networks," in *Proc. Allerton Conf. Comm., Control*, pp. 204-211, Urbana, IL, Oct. 2017.

[3] S. Dhamal, "Effectiveness of Diffusing Information through a Social Network in Multiple Phases," *IEEE Globecom*, pp. 1-7, Dec. 2018.

[4] T. -H. Fan and K. -C. Chen, "A New Social Network Model of Online Forums," *IEEE Globecom*, pp. 1-6, Dec. 2017.

[5] I. Srba, M. Bielikova, "A Comprehensive Survey and Classification of Approaches for Community Question Answering," *ACM Trans. on the Web*, Vol. 10, No. 3, pp. 1-63, Aug. 2016.

[6] https://quora.com/ and https://stackexchange.com/

[7] I. Ali, R. Chang, J. . Chuang, C. Hsu and C. Yetis, "Optimal Question Answering Routing in Dynamic Online Social Networks," *Proc. IEEE Vehic. Tech. Conf.*, pp. 1-7, Sep. 2017.

[8] P. Kumar and S. Meyn, "Stability of Queueing Networks and Scheduling Policies," *IEEE Trans. Aut. Con.*, vol. 40, pp. 251-260, Feb. 1995.

[9] P. Bremaud, Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues, Springer, 2008.

[10] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge.

[11] H. Nassif, M. Mohtarami, and J. Glass, "Learning Semantic Relatedness in Community Question Answering Using Neural Models", *Proc. 1st Workshop on Representation Learning for NLP*, pp. 137–147, Aug. 2016.