WHO SHOULD ANSWER MY QUESTION? QUESTION ROUTING IN Q&A WEBSITES.

Final report for presented for the degree of Master 2 Sciences des Systèmes Complexes

July 31, 2016

David Pardo

Supervisors: Nidhi Hegde — Bell Labs researcher
Laurent Viennot — INRIA researcher

ÉCOLE
POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Who should answer my question? Question routing in Q&A websites.

David Pardo[1]

[1]École Polytechnique, Palaiseau,France
Microsoft Research-INRIA Joint Centre, Palaiseau, France
Laboratory of Information, Networking and Communication Sciences, Paris, France
Email: davidpb90@gmail.com

The problem of question routing in Q&A websites has become highly relevant in the recent years. Most of the current studies tackle the problem from an expert finding perspective. Focusing on expertise can overload the most active responders while ignoring users who might have enough knowledge to solve certain questions. We propose a novel perspective to address the problem, i.e. a multistep routing framework. Different methods are part of the introduced framework, whose properties are studied using data from the Q&A website Math StackExchange. In particular some geometric properties and its user loading features. We finish by integrating the different methods considered into the proposed scheme and analyzing its behavior in the data, with the help of an data driven oracle. The empirical study shows the potential benefits of implementing this framework in order to increase the fraction of solved questions.

1. Introduction

Nowadays most of our interactions with information sources happen online. In particular, it is common to surf the web for addressing questions in a daily basis. Thus, question and answer (Q&A) websites have become a very useful tool for obtaining answers in specific topics in a fast an easy way. Most of these sites are based on free community sharing, with users being able to ask and answer questions with minor restrictions. This creates complex ecosystems with rich interactions where subtopics and expert users emerge naturally [1]–[6].

We are interested in a very specific yet fundamental problem in Q&A websites: given a new question, how can we find a user who can answer it appropriately? Some of the most popular sites (e.g. StackExchange, Quora, Yahoo! Answers) trust in the collective crowds intelligence, by posting lists of unanswered questions that all users can access and attempt to answer. This lists can be filtered by a user by selecting particular topics, words or askers of interest. However, they do not have in place algorithms for finding potential responders for a new question.

Personalizing the websites might be deemed too expensive, but we argue that it would be important from a user perspective. In Figure 1 we observe that for the Math StackExchange site more than 20% of the questions do not get an answer in the first two days and more than 30% do not get an up voted answer in the same period. As the curves suggests, most of the questions get answers in the first 4 hours, and due to the unanswered questions list systems, it is likely that after two days without an answer a question will never get one [7]. Similar results have been found for other sites, including StackOverflow, the most popular within this family [8]. This shows that algorithms for automatically directing questions to possible responders have a big potential in terms of both user satisfaction and growth of the solved questions pool.

Finding appropriate responders has become a hot research topic in the last years [9]–[20], with most of the studies focusing on expert finding [21]–[27]. This implies finding users with proven expertise in a given topic in order to solve a proposed question. By using network analysis, probabilistic topic models or a combination of both, users are ranked according to their probability of answering. This

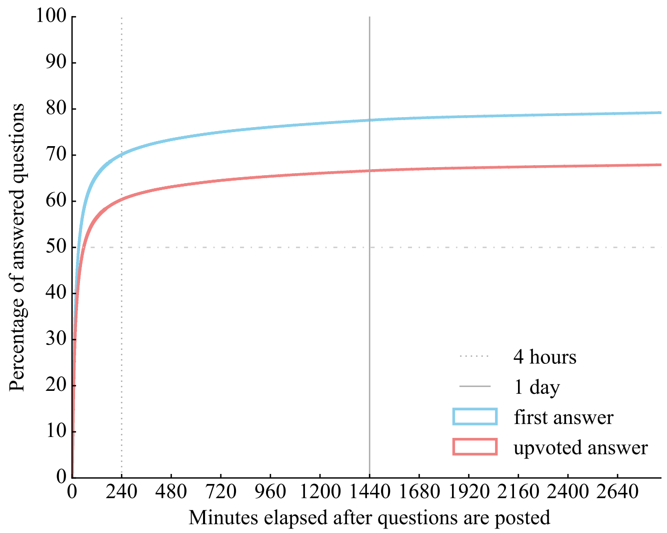

Fig. 1: Cumulative density of the time elapsed before questions receive answers. It is observed that almost a third of the questions do not receive an upvoted answer, thus justifying the development of question routing methods.

process involves having a pool of solved questions from which the models can be learned. Afterwards, with the inferred parameters it is possible to predict potential users for the incoming questions.

In this paper we want to introduce a new paradigm, to the best of the author's knowledge, that is able to solve the problem independently, but which could also be used as a complement to existing approaches. Instead of focusing solely on a given history of user interactions, we argue that a routing approach can be beneficial. We propose sending a new question to one user (or a small subset) and, in case she does not answer, use her feedback and other information that we might have to route the question towards another user, iterating this process until a responder is found. In other words, we want to send the question through a social flow where new information is constantly added until we find someone who can solve it. This approach is related with reinforcement learning [28], since we are using an initial pool of solved questions to find a first candidate, but then the system tries to learn new information from each potential answerer on the go.

Apart from representing a completely new approach worth exploring on its own, this proposal can tackle situations where the historical data available is not good enough for inferring models that can provide strong results. Moreover, by not doing an explicit search for experts, it has the potential of alleviating problems related with user overload when the number of experts is low. We provide our own methods for finding the first pool of candidate answerers, with simplicity and scalability being our main concerns. Nevertheless, depending on the specific task, this methods can potentially be replaced by approaches with more predictive power such as probabilistic topic modeling, which combined with the routing approach would increase their efficiency. In view of this, we want to stress that our main contribution is the general framework of multistep routing and its versatility.

The remaining of paper is organized as follows: In Section 2 we discuss with more detail the related literateure in this topic. Then, in Section 3 we formally present the algorithmic framework we propose. Afterwards, Section 4 deals with the use of users' histories to find potential candidates, including a geometrical analysis of the underlying space via the doubling property. Section 5 introduces an approach for finding possible responders via neighboring questions. Subsequently in Section 6 we present an algorithm based on the ideas introduced on previous sections and show how it performs in the analyzed data. We finalize with some concluding remarks and perspectives in Section 7.

A. The Data

For the present work we used public data provided by the family of Q&A websites StackExchange (https://archive.org/details/stackexchange). Each website in this family has a specific topic, with Computer Programming (StackOverflow) and Mathematics (Math StackExchange) being the most popular ones. This data was selected for sundry reasons:

- These websites contain all the basic functions of general Q&A sites, which can be explored without the need of dealing with more specific features of this family.
- StackExchange is one of the most successful approaches to Q&A websites [8].

- Several related studies have been performed and validated using these data, e.g. [7], [8], [16], [29]–[31].
- Updated versions are publicly available in appropriate formats.
- The data dumps provide all the necessary information of users and posts for performing in depth analysis.

Specifically we used a subset of the Math StackExchange data dump from December 2015, consisting of the first 65138 questions, their corresponding 125233 answers and 254765 comments, provided by 13658 users. The data, originally in XML format, were transformed into SQL tables using code made available by [8]. All the analyses were performed in Python 2.7, with the main data handling and analysis tasks being done using Pandas library.

2. Related Work

Previous studies have tackled this problem from a very natural perspective: finding experts. Expertise estimation and prediction is a problem in its own right with several studies dedicated exclusively to it [4], [6], [32], [33]. Regarding the search of potential responders, the general idea is to identify the topic of the given question and rank the users in terms of their expertise in the given topic. The highest ranked users would be the candidate responders. Two main approaches have been used: link analysis and topic modeling.

A. Link analysis

Different natural graph structures can be extracted from the interactions in Q&A websites. For example a directed graph between users according to who has answered whom, a bi-layered graph of questions and users, or a bi-layered graph of posts and users. In all cases it is possible to identify authority rankings for users using different techniques.

[9] proposed a HITS-like [23] algorithm for identifying expert users in Yahoo! Answers. [25] evaluated different expert ranking algorithms, including a novel ”ExpertiseRank” inspired on PageRank [34]. A combination of HITS and PageRank algorithms was introduced by [24] to identify an unspecified number of experts in Yahoo! Answers. [35] introduced a hybrid method using link analysis with language modeling to find experts within the target category for a given question. Lastly, [10] developed a method to detect the relevant categories for a given question and find experts using a graph built upon this information.

All these methods have one limitation in common: Given a new question, they assign the most expert users in the relevant topic as the potential responder. This can create an over load on the experts, since all questions in a given topic will be sent to them. However, less expert users are likely to be able to answer low and medium level questions, thus the overload could be overcome.

B. Topic Models

The textual content of questions and answers is used to train latent topic models [36] that can in turn predict the user with the highest probability of answering a given question. Additionally, other specific features of Q&A websites have also been used, e.g. tags and votes.

[11] introduced the probabilistic topic model approach, they identify topics using LDA [37], train a graphical model based on user history and finally use a topic sensitive PageRank-like algorithm (TSPR) to find the expert users likely to answer a given question. [13] independently proposed a similar method for accounting for topic relationships among users. [12] identifies latent question topics via pLSA [38] and then finds the correspondent experts. Best possible responders are ranked using a probabilistic language model by [39]. [15] improves on the previous models by separating the users modeling according to their roles as askers and responders. [14] finds topic specific influential users in Twitter by leveraging different topic models. [40] introduced Topic-level Expert Learning, incorporating link analysis into content analysis, thus improving upon previous models in terms of expert finding efficiency according to various metrics. Finally, [16] combines link analysis with a topic-expertise model which learns user and topic specific expertise levels.

The aforementioned models provide a more diverse pool of experts due to its focus on topic identification and, in some cases, expertise level distinction. Nevertheless, generality is sacrificed, since specific features of the websites being analyzed are normally used. Moreover, they rely almost completely on the textual information provided in the questions and answers. However, multimedia content is becoming ubiquitous nowadays, thus some knowledge sharing sites might not have readily available texts to learn the models efficiently.

3. Routing Algorithm

We propose a novel and simpler method to find users who can answer a given question, i.e. routing, inspired in graph studies, mainly focused on the internet [41]–[44]. We aim for simplicity in order to develop a general scalable framework in which more complex models might be integrated depending on specific goals. Furthermore, although inspired in Q&A websites, this framework can be extended to other task assignment problems by modifying the methods involved in the different stages. In contrast with previous works, it entails a multistep approach:

- A new question $\mathbf{q}$ is posted by a user $\mathbf{a_q}$, the asker.
- Using the information given by $\mathbf{q}$ and $\mathbf{a_q}$, a candidate responder $\mathbf{c_{1,q}}$ is selected.
- If $\mathbf{c_{1,q}}$ refuses to answer, it is added to a set of refusers $\mathbf{R_q}$ whose information is used to find another candidate $\mathbf{c_{2,q}}$.
- This process is performed recursively until a candidate $\mathbf{c_{n,q}}$ answers the question, i.e. until a responder $\mathbf{r_q}$ is found.

This process can be presented more formally in an algorithmic way:

Algorithm 1: An algorithmic principle for routing a new question $\mathbf{q}$ in order to find a responder.

```
function AlgorithmicPrinciple $(\mathbf{q}, \mathbf{a_q}, \mathbf{R_q})$.
Input  : A question $\mathbf{q}$, an asker $\mathbf{a_q}$, a set $\mathbf{R_q}$
         of users refusing to answer, originally
         empty.
Output: An answer for $\mathbf{q}$.
Main call: function FindAnswer($\mathbf{q}$,$\mathbf{a_q}$, $\varnothing$).
Recursive call in function Find($\mathbf{q}$,$\mathbf{a_q}$,$\mathbf{R_q}$).
    Find candidate responder $\mathbf{c_{i,q}}$ using $\mathbf{q}$,$\mathbf{a_q}$
    and $\mathbf{R_q}$.
    If $\mathbf{c_{1,q}}$ does not answer then
    Find($\mathbf{q}$,$\mathbf{a_q}$,$\mathbf{R_q} \cup \mathbf{c_{1,q}}$).
```

The main open problem consists on finding the potential responders using the initial data and the information provided by each new refuser. When a new question is provided, two sources of information are available: the asker and the question itself. We explored each of them separately in order to delve into the possibilities that they provide on their own.

4. User history

A. Basic definitions

The history of interactions of a user in the site stores valuable information of the kind of questions she is interested in and of other users with shared interests. Let us define a set of users $U = \{u_1, u_2, \dots, u_n\}$ and a set of questions $Q = \{q_1, q_2, \dots, q_m\}$. We define a bipartite graph $G = (V, E)$, with a set of vertices $V = V_u \cup V_q$, composed of two layers, one of users V_u and one of questions V_q, which are connected by three different kind of edges $E = E_a \cup E_r \cup E_c$. An edge $e_a \in E_a$ connects a user with a question she has asked, $e_r \in E_r$ links a user with a question she has answered, and finally $e_c \in E_c$ unites a user with a question she has commented in. Figure 2 illustrates the aforementioned network.

This natural graph structure permit us embed the data in a discrete multidimensional space. In order to explore its structure and its usefulness to the problem in hand, we define two sets of interests for each user u and two sets of users with common interests:

- *Definition 1.* We define the *reduced set of interests* of u as $S_u = \{q \in Q : \exists\ e \in E_a \cup E_r$ that connects u with $q\}$. In other words this is the set of question which a user has either asked or answered. This would correspond to question in the topics with more interest

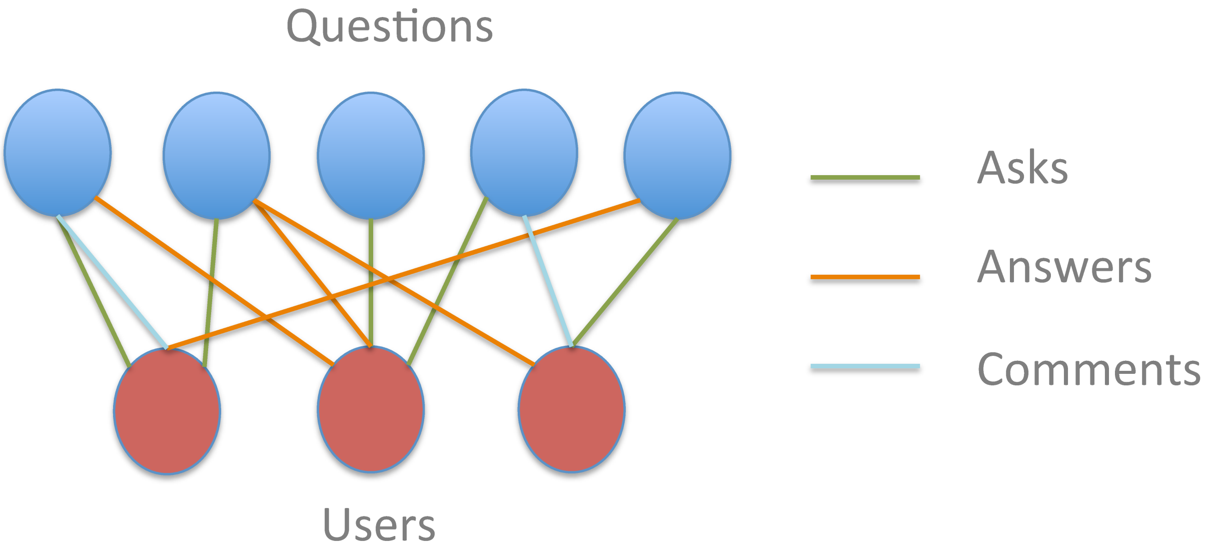

Fig. 2: Graph structure emerging from user interaction in Q&A websites. It can be understood as a bilayered network with three different kinds of links between the groups.

for the user and where she has some kind of expertise.

- *Definition 2.* The *large set of interests* of u is $L_u = \{q \in Q : \exists\ e \in E_a \cup E_r \cup E_c$ that connects u with $q\}$. These corresponds to all the questions in which a user has interacted, thus representing her broader interests, including topics in which she might have no expertise whatsoever.

- *Definition 3.* The *reduced set of users with common interests* for u is defined as $RC_u = \{u_i \in U : L_u \cap R_{u_i} \neq \varnothing\}$.
- *Definition 4.* The *large set of users with common interests* for u is defined as $LC_u = \{u_i \in U : L_u \cap L_{u_i} \neq \varnothing\}$.

B. Doubling property

Definition 5. Let X be a metric space. Let α_r be the number of balls B_r of radius r needed to cover a ball B_{2r}. If α_r is bounded, then X is said to meet the *doubling property* [45]–[47]

and the least upper bound α is called the *doubling dimension* of X.

A low doubling dimension have been shown to be a desirable property for routing algorithms in general metric spaces [44], [47]–[49]. Hence we want to explore a similar concept for the discrete space we defined above.

Definition 6. Let $u \in U$ with large set of interests S_u. We define a *cover* of u to be a set of users $C_u \subset U$, such that $L_u \subset S_{u_1} \cup S_{u_2} \cup ... \cup S_{u_m}$. The size of the smallest cover for u is called its *cover number*.

We observe that the cover number can be interpreted as an analog of α_r for our space of interests. Hence, we explore whether this cover number is bounded, which would imply a potential good behavior of routing algorithms.

Figure 3 shows the cover number of u as a function of the size of L_u. This plot illustrates that for most of the users the cover number is bounded by 10, with less than 1% having a larger one. Moreover, the cover number is calculated using a greedy algorithm, i.e. it is an approximation where not necessarily the least upper bound is found. This two facts indicate that the cover number is likely bounded and that even if it is not the case, the covers might be used to efficiently find responders for a considerable majority of the questions.

C. Finding potential responders

The results shown above suggest a potential algorithm consisting on using the asker's cover as a potential set of candidates. However, there are two risks involved in this procedure, namely:

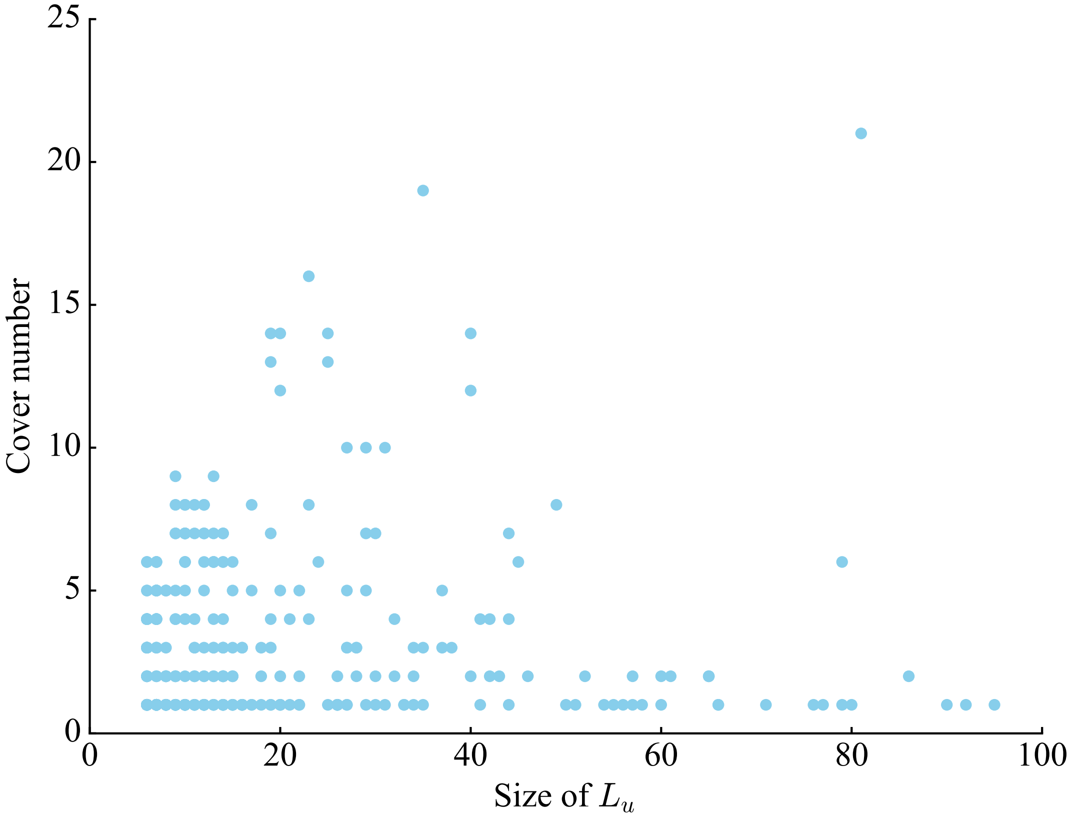

Fig. 3: Cover number vs $|L_u|$ for all $u \in U$. Although the cover number is relatively big for some users, most of them have a number lower than 10. Furthermore, there is no clear incremental tendency for this number, thus a bounded cover number cannot be discarded for this kind of websites.

- Given that the algorithm for finding the cover is greedy, it is likely that it will select a limited number of users multiple number of times, those who are more active on the site. This would generate overload on these users, with the algorithm becoming an expert finder, if expertise is understood as high activity.

- Two users can share the same interest with none of them being knowledgeable enough to answer a question. Some very active users might be just learning about a topic, i.e. just asking questions, and yet the greedy algorithm would select them as potential responders.

In view of this, we decided to consider the whole sets of common interests and to investigate their behavior as sources of potential responders. We take 90% of the questions selected at random with their respective askers as a training set Q_{train} and the remaining 10% as a test set Q_{test}. Given $q, u \in Q_{test}$ we want to answer two questions about the sets RC_u and LC_u:

- Do they contain a real answerer of q?
- How big are they?

First of all, we can observe in Figure 4 that above 90% of the questions have less than 4 answerers, which put into perspective the task in hand, i.e. for most questions we have at most 4 out of over 13000 users that we want to find. In Figure 5 we observe the cumulative histogram of the number of real responders found in the reduced set of common interests RC and in the large set of common interests LC. The results for LC are evidently better, with at least one responder found for around 95% of the questions and finding multiple ones in half of the cases.

However, it is expected that a larger set have more success in having members corresponding to real responders. To obtain the whole picture we need to see how much these sets differ in

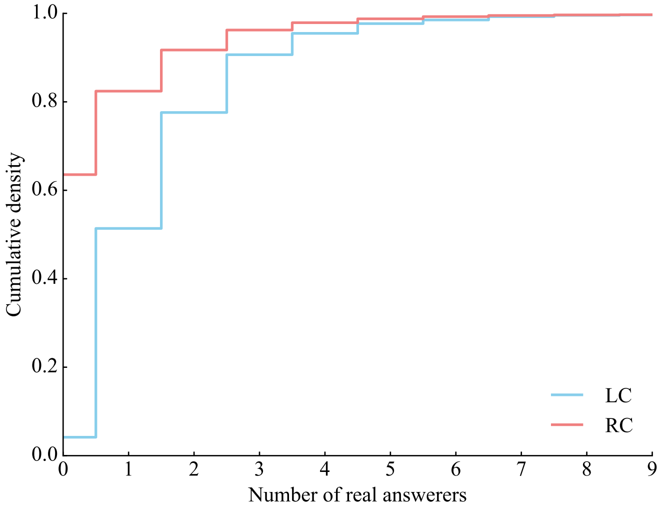

Fig. 5: Cumulative density of the number of real responders found in LC_u and RC_u. By definition the results for LC_u have to be better, however the difference is striking. By exploring LC it is possible to find a responder for about 95% of the questions.

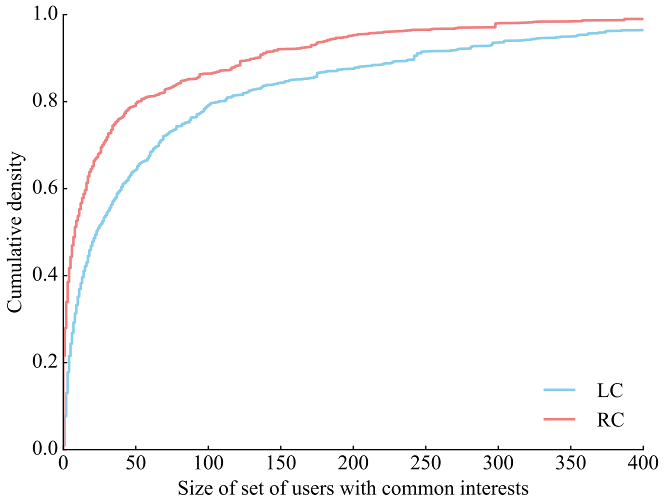

Fig. 6: Cumulative density of $|LC_u|$ and $|RC_u|$. The tradeoff between recall and the number of potential candidates to consider is illustrated. Considering LC obliges to browse about twice the number of users. Nevertheless, the sacrifice pays off in this case, since the recall more than doubles, as illustrated in the figure to the left.

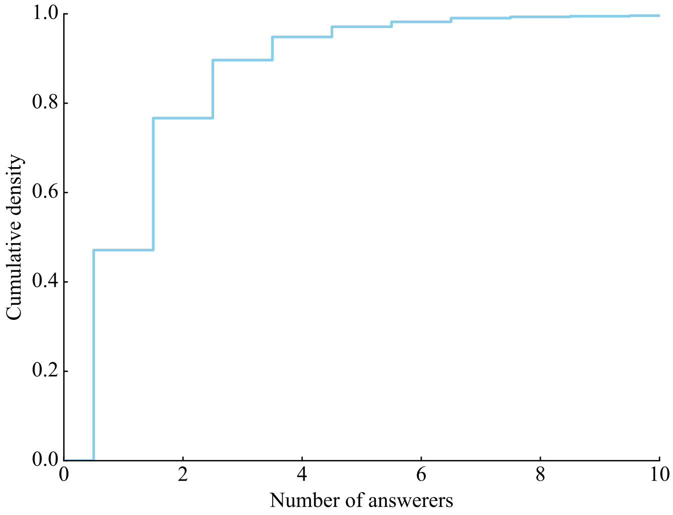

Fig. 4: Cumulative density of the number of responders per question. It is seen that above 90% of the questions have less than 4 answerers.

size. Figure 6 illustrates the cumulative histogram of sizes for both sets. In general, for a given percentage of users LC_u tends to be double the size of RC_u, e.g. for 80% of the users $|RC_u| < 60$, whereas to obtain the same percentage with $|LC_u|$ we need to consider sizes up to 120.

Hitherto we have not observed anything entirely surprising, since a compromise between accuracy and set size is habitual. Nonetheless, considering random sets of users of the same size as the ones shown above yields less than 3% of successful cases. This recall clearly illustrates the striking advantages deriving from evaluating user's histories. It remains to find tools to reduce the set of candidates and to tackle the problem of finding an answer for the small subset of question for which an answerer is not present within the historical record of the asker.

Before addressing this issue we analyze the load on users deriving from considering LC as the source of answerers. Figure 7 illustrates the frequency of answers per user, i.e. how many times a user was found to be the real responder of a given question when being part of the set of common interests of the asker. Although there exists overloading for a handful of users, the distribution is left-skewed and with a light tail. Thus the plot shows a good task repartition, in that most of the users are selected to answer a few questions, hence creating a dynamic social interaction not dependent on a small set of experts.

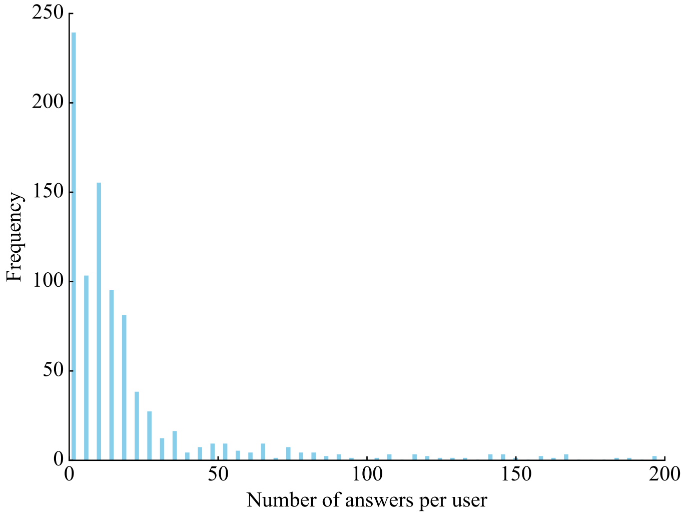

Fig. 7: Histogram of the number of answers given per user. Although there exists a small number of highly requested users, there is a considerable number of users providing answers, suggesting a rich ecosystem not necessarily subordinated to experts.

5. Similar questions

A. *Defining a metric*

The question themselves, the texts, contain all the information available about their topic and their difficulty. We want to automatically extract this information with the goal of finding an appropriate responder. Moreover, if we find relations between questions, we can also use the historical information implicit in a new question, i.e. which users have answered similar questions. Finding similar question is an interesting problem in itself, an advance techniques exist to tackle it [16]. However, once again we opt for simplicity, using distances between questions to define their degree of similarity.

The most common technique for embedding texts in a metric space, the so called Vector Space Model [50] developed in the file of information retrieval, consists on creating a weighted frequency matrix of words in documents [51], [52] in which each text is represented by a vector. However, there are two main problems with this approach: it is not scalable since the matrix grows with the number of texts and the number of words, the matrix having to be recalculated every time a new text is considered, since the frequencies of words across the entire corpus are modified. Hence, we decide to use a metric that can be calculated by means of the two questions of interest only. This means that we do not explicitly build a metric space, i.e. we only have the relative distances between questions, but not their exact location.

We use the Joint Complexity [53] metric, a state of the art method which has been recently used to successfully cluster tweets [54], which validates it as a good information retrieval tool for short texts.

Definition 7. Given a sequence X, let $I(X)$ be its set of factors, for example if $X =$'tree', then $I(X) = \{t, r, e, tr, re, ee, the, see, tree, \varnothing\}$. The *complexity* of X is defined as its number of factors, i.e. $|I(X)|$. The *joint complexity* of two sequences X and Y is defined as the number of common factors, i.e. $JC(X,Y) = |I(X) \cap I(Y)|$. A similarity between texts is defined as:

$$s(X,Y) = \frac{JC(X,Y)}{|I(X)| + |I(Y)| - JC(X,Y)}. \quad (1)$$

Finally, a distance can be defined as the inverse of this similarity, where $d(X,Y) := \infty$ when $s(X,Y) = 0$.

With this approach, if we have a new question, we would only need to calculate its distance to any other question of interest, while for the remaining questions no changes are necessary. This implies a linear scaling with the number of total questions, which suits our goal of developing a scalable algorithm. Nonetheless, It is important to remark that we are sacrificing the important semantic properties of text, since we are focusing only in statistical properties which do not relate to the meaning of the words whatsoever.

B. *Finding potential responders*

Given a new question q, we define a simple procedure to select potential responders:

1) Find $d(q, q_{old})$ for all existing questions q_{old}.
2) Sort the questions according to $d(q, q_old)$.

3) Select as potential responders those users who had answered the nearest questions to q.

We define the same training and test sets as in the previous section. Given a question $q \in Q_{test}$ we calculate the index i of the first question $q_i \in Q_{train}$ for which $R_q \cap R_{q_i} \neq \varnothing$, i.e. the first question for which there is a responder in common. In Figure 8 we observe that for over 40% of the questions we obtain a positive result within the first 50 neighbors, i.e. evaluating less than 0.5% of the training set. Although the percentage might seem low compared to the real percentages of answered questions, we have to take into account that we are using only the information provided by the text and using a scalable metric that does not evaluate relations across the whole corpus. Moreover, the performance boost with respect to a random selection of subsets of questions with the same size confirms that the metric is having a considerable positive effect.

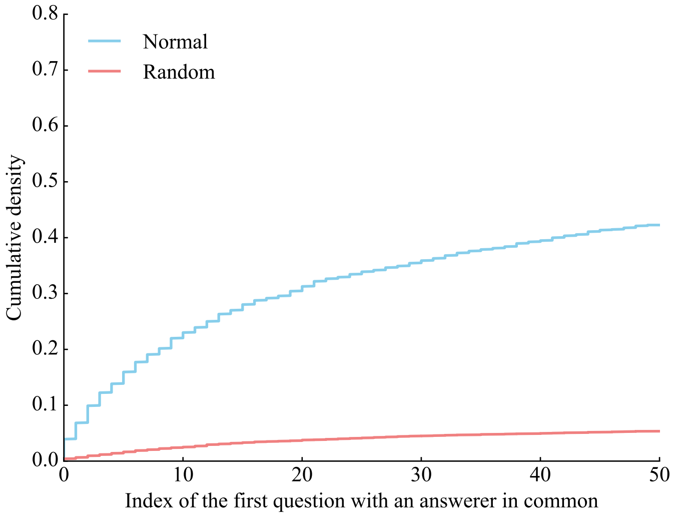

Fig. 8: Cumulative density of the index of the first question with a responder in common with a given question. It

Therefore, this approach shows great potential as a tool to implement in a question routing algorithm. Moreover, it is remarkable that the results are slightly better when using only the question title, which suggests that this metric is better suited for short texts, with additional textual information being rendered negligible. This fact also signals to this method's potential for other task assignment problems, e.g. we can imagine a Q&A website based on videos, where the only readily available information would be the title of the question video, or services like Amazon's Mechanical Turk where micro jobs are offered to subscribers, often with concise descriptions.

As a final remark, the user load for this method presents the same general features as for the graph approach, as shown in Figure 9. A few users answer a big number of questions, but in general questions are distributed among several responders. Hence, both approaches considered were found to avoid overloading problems, as desired.

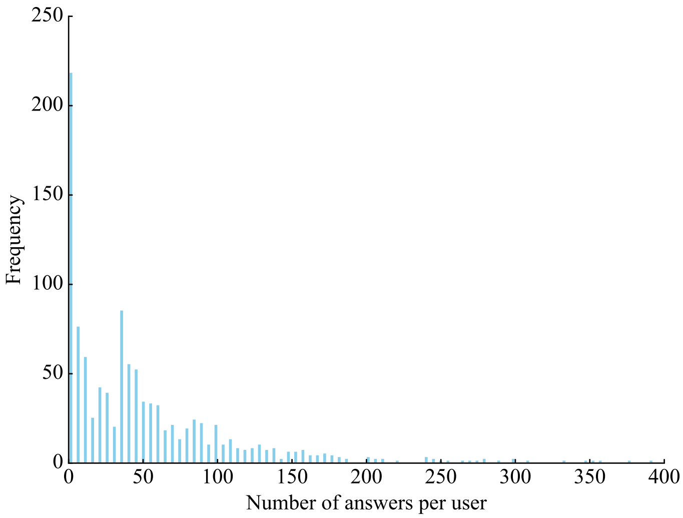

Fig. 9: Histogram of the number of answers given per user. The same general features as in the graph approach are observed, in particular the avoidance of user overloading.

6. Question Routing Algorithms

We first propose an algorithm solely based on the information provided by the question. It basically consists on ordering the existing solved questions according to their distance to the new one $\mathbf{q_n}$, selecting one of their answerers as a candidate responder, and iterating over these candidates until an answer is given. Additionally, in order to use the refusers' feedback to optimize the process, we consider two subsets of questions: Related Questions $\mathbf{RQ}$, consisting of $\mathbf{q_n}$ and other questions that refusers consider related; and Unrelated Questions $\mathbf{UQ}$, including all questions that refusers deem unrelated. When a question $\mathbf{q}$ is considered, a candidate will be selected from its set of responders only if $\mathbf{q}$ closer to $\mathbf{RQ}$ than to $\mathbf{UQ}$. On the contrary, $\mathbf{q}$ is included in $\mathbf{UQ}$ without querying any user in its set of answerers.

Algorithm 2: An algorithm for routing a new question $\mathbf{q_n}$ based only on its text.

```
function RoutingQuestion($\mathbf{q_n}, \mathbf{Q}, \mathbf{UQ}, \mathbf{RQ}$).
Input  : A question $\mathbf{q_n}$, the set of solved
         questions $\mathbf{Q}$, a set of unrelated
         questions $\mathbf{UQ}$ originally empty and a
         set of related questions $\mathbf{RQ}$ originally
         containing $\mathbf{q_n}$
Output: An answer for $\mathbf{q_n}$
for q $\in$ Q do
    Calculate $d(\mathbf{q_n}, \mathbf{q})$.
    Sort $\mathbf{Q}$ according to $d(\mathbf{q_n}, \mathbf{q})$.
end
while responder is not found do
    for q $\in$ $\mathrm{Q_{sort}}$ do
        if $d(\mathrm{q}, \mathbf{RQ}) > d(\mathrm{q}, \mathbf{UQ}) - \alpha$ then
            $\mathbf{RQ} = \mathbf{RQ} \cap \mathbf{q}$.
        else
            Select a responder $\mathbf{r}$ of $\mathbf{q}$, who will
            either: answer $\mathbf{q_n}$ or say whether it
            is related or unrelated to $\mathbf{q}$.
            if $\mathbf{r}$ solves the question then
                return answer.
            else if $\mathbf{r}$ says $\mathbf{q}$ is related to $\mathbf{q_n}$
            then
                $\mathbf{RQ} = \mathbf{RQ} \cap \mathbf{q}$.
            else
                $\mathbf{UQ} = \mathbf{UQ} \cap \mathbf{q}$.
            end
        end
    end
end
```

Given that the Joint Complexity metric does not capture semantic information, we add a forgiving parameter α with the goal of avoiding a big quantity of false negatives. Questions for which the difference between its distance to $\mathbf{RQ}$ and $\mathbf{UQ}$ are thus still considered useful.

Having two complementary approaches for finding potential responders, we will integrate them into a routing scheme. We propose a mixed algorithm considering both the question's information and the asker's history. We use the asker's information to select a subset of questions to be evaluated by the algorithm illustrated above. Briefly stated, we use LI_a as an input to **RoutingQuestion** and in case an answer is not found we move into the remaining questions.

Algorithm 3: An algorithm for routing a new question $\mathbf{q_n}$ based on its text and its asker's history.

```
function RoutingQuestionUser($\mathbf{q_n}, \mathbf{a_q}, \mathbf{Q}$).
Input  : A question $\mathbf{q_n}$, its asker $\mathbf{a_q}$ and the
         set of solved questions $\mathbf{Q}$.
Output: An answer for $\mathbf{q_n}$
Find $\mathbf{LI}(\mathbf{a_q})$.
RoutingQuestion($\mathbf{q_n}, \mathbf{LI}(\mathbf{a_q}), \varnothing, \mathbf{q_n}$).
if $\mathbf{q_n}$ is not solved then
    RoutingQuestion($\mathbf{q_n}, \mathbf{Q} \setminus \mathbf{LI}(\mathbf{a_q}), \varnothing, \mathbf{q_n}$).
end
```

A. Validation

Since multistep routing mechanisms are not implemented in existing Q&A websites, we propose an oracle to validate the proposed algorithm with StackExchange's data:

- Given a question $\mathbf{q}$, select a question $\mathbf{n_q}$ for finding potential users according to Algorithm 3.

- Query a user $\mathbf{u_n}$ who has participated in $\mathbf{n_q}$. This is where the oracle performs its prediction: we consider that a question was deemed useful by $\mathbf{u_n}$ if one of the users interacting with $\mathbf{n_q}$ commented in $\mathbf{q}$. If one of such users actually had answered $\mathbf{q}$, we consider that we found a responder.

- Proceed as indicated in Algorithm 3 to select the next candidate.

Figure 10 shows the cumulative density of the number of real responders found for different procedures. Let us dissect the results:

- Normal: No questions are skipped, i.e. it is the base case in which no feedback is given.
- Algorithm: The algorithm as presented above, with $\alpha = 0.001$.
- Smart algorithm: The same algorithm, but in a positive scenario in which questions with real responders are not skipped.
- Best: Best case scenario in that all questions in which a refuser has participated are ignored, except in the case of a real responder being present.
- Random: Instead of considering the questions ordered by distance, we take a set of random question, with the same size as the initial pool of questions considered.

Given that we are using an ad hoc oracle, we also consider different possible performances for our proposed algorithm. We observed that the no feedback case performs better than the basic algorithm, however both the smart version and the best care scenario reveal the cause. It is not the case that the algorithm does not achieve the purpose of reducing the number of questions to query, but that under the oracle used it often skips the fist question with a responder in common. In fact, a limited pool of the 300 hundred nearest questions was considered for the analysis and yet we find that the algorithm runs through all the questions in only 5 steps. This shows its great potential for reducing routing times. In a real implementation, even when useful questions are skipped, the algorithm would reevaluate them after exploring the whole dataset, or would find a successful question further down the road.

There are some important things to remark: even under a non optimal oracle, the smart and best case approaches show the potential benefits of routing in terms of reducing the number of required steps. Furthermore, the difference between all versions of the algorithm and the random case illustrates that the methods put in place to select candidate responders are effectively accomplishing their goal. Additional confirmation comes from looking at the high percentage of questions for which a responder is found in the very first question. This fact shows the power of the Joint Complexity measure for finding related questions by just analyzing their titles.

With real user feedback these results would quite likely show a bigger difference in performance when using the information at each step. Evidently more data is necessary to support these claims, but given that with the simplest possible feedback our algorithm shows potential benefits, it is not a stretch to think that with a multistep routing system with appropriate and more extensive feedback put in place, the performance of the method would noticeably increase.

B. Analysis

The integration of Algorithm 2 into Algorithm 3 represents the heart of our proposal. We simply require a procedure for having a first pool of candidates and a way to classify them. Afterwards, due to the nature of this routing proposal, it is guaranteed that we find an answerer insofar there exists one. Of course this is trivially true, since in the worst case scenario we will ask all the users. Thus, our main concern is reducing the number of steps required to find such user.

As shown, the idea of using the refusers' feedback can achieve the desired decrease, with the added advantage that it can be a complement of already existing systems. For example, taking the case of StackOverflow, the main mechanism could remain to be the trust in the extensive activity of users, with the reward system put in place, since it is quite efficient. But with that fraction of questions that don't get an answer, about a fifth, a method such as the one we present can be applied, instead of condemning these questions to oblivion. Taking into account that the questions are likely to be forgotten at the end of the list, it is not a problem to wait for the feedback of some users, taking into account that this might actually reactivate an agonizing question.

This implies that recall is not a concern for this method. In terms of time, the only complex tasks to perform are the calculation of the Joint

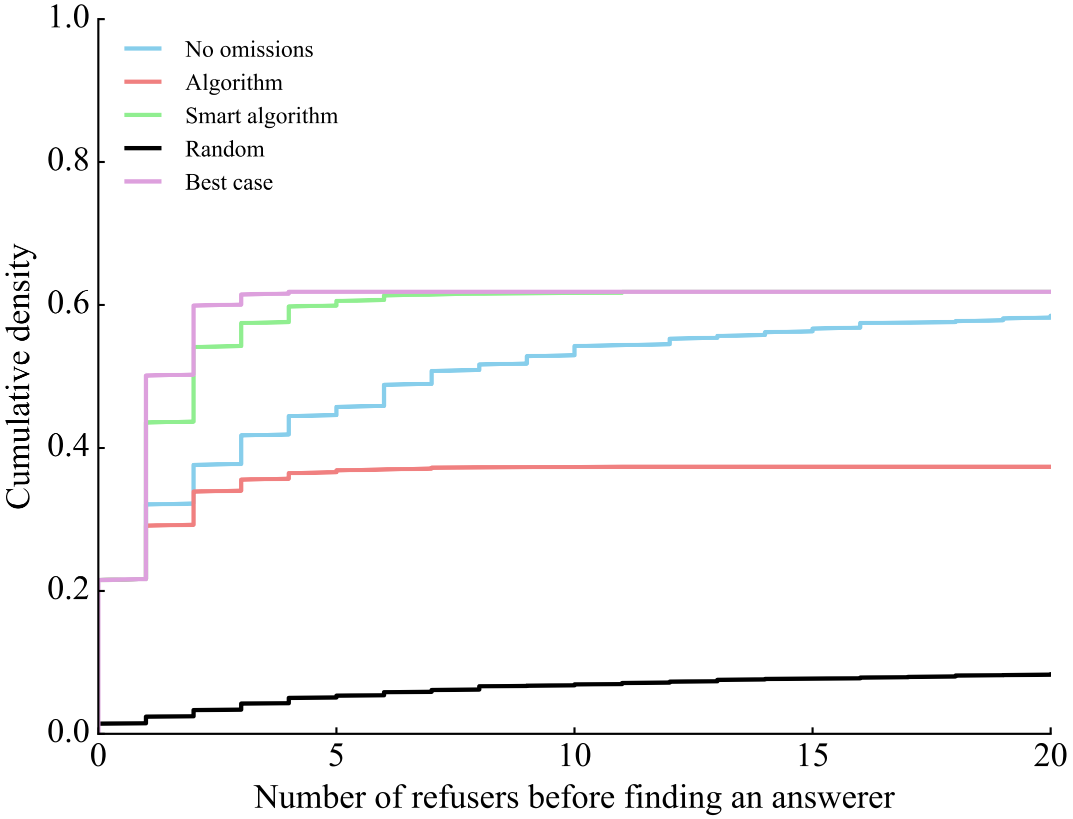

Fig. 10: Cumulative histogram of users queried before finding a responder under different methods. Although the basic algorithm underperforms with respect to the no feedback case, the two improved versions show the potential that good feedback has for significantly improving the routing mechanism.

Complexity distances between the question and the pool of existing questions and the sorting of the pool according to such metric. Both of these processes, under appropriate implementations, have computational complexity of $O(n \log n)$ in the worst case [53], [55], which makes the method scalable. However, because of its multistep nature we cannot consider specific timings for a given question without real user interaction.

It is not to be forgotten, though, that our aim is to offer a framework where other efficient solution can be integrated and where more specific data can be used, rather than replace and improve upon existing algorithms in terms of efficiency in expert finding tasks. Scalability and avoiding user overload were important properties that were shown to be met. Versatility and generality are also features of our approach, since any of the steps can be adapted to work with different methods or more features. For instance, it is possible to replace the method for finding the first pool of candidates with a state of the art topic model which would provide us at the same time with an alternative metric to be used in the routing task. Moreover, it is also imaginable to think of different kinds of user feedback, such as voting, rating or recommending a proposed question, in which case more complex metrics would be required to make use of the additional information.

7. Conclusions and Future Work

Our main contribution is the proposed scalable and general framework for integrating responder finding methods into a routing scheme for finding appropriate responders in Q&A websites and more generally in task assignment problems. We also proposed our own tools for finding candidate responders. Defining a natural graph structure and sets of interests for the users, we found that the cover number is bounded for a considerable majority of users, which suggests that an answer can be found in a reduced number of steps via routing.

Moreover, we found that the large set of interest of a user is an effective pool for finding potential responders. This fact is a strong evidence that asker's history is as important or even more so than the question itself. On the other hand, embedding the questions in a metric space was shown to be a potential complement for graph techniques, although the results obtained via the Joint Complexity metric were less definitive than those yielded when considering the user graph.

Novel algorithms based on a routing scheme were introduced. Its potential to find answers to a new question in a low number of steps was shown using oracles based on answerers and commenters for the questions in MathExchange's dataset. However, further studies are required to measure the degree of positive impact of feedback on the overall process. Furthermore, due to the general features of the algorithms, they can be used in other related applications such as video forums or in general in the broader context of task assignment.

The proposal of new text metrics would be an interesting problem with a lot of potential, specially if they are integrated into models that can take into account other information sources such as ratings and votes. Additionally, it remains to integrate other state of the art methods into the proposed framework, such as probabilistic topic models.

The geometrical properties of both the discrete space and the metric space are to be further analyzed, since the present results, although promising, were not totally conclusive in terms of the boundedness of the cover number. The possible connections between the doubling property of general metric spaces and the cover number introduced in this work can be studied in more detail.

Finally, a major goal is to implement these algorithms in real applications , with the additional benefit that new data with information about feedback interactions would be obtained and released.

Acknowledgements

I am really grateful to my supervisors Nidhi Hegde and Laurent Viennot for their constant support and insightful ideas. I also want to thank the feedback given by colleagues and professors, spatially to Khashayar Pakdaman. Finally, I want to express my gratitude towards the people involved in the Centre of Complexity Science and the Erasmus Mundus Program for the opportunities that I have received throughout these years.

This work was funded by Microsoft Research-Inria Joint Centre, located in Saclay, France, being part of my master studies sponsored by the Education, Audiovisual and Culture Executive Agency of the European Commission. I was hosted by the Laboratory of Information, Networking and Communication Sciences in Paris.

References

[1] H. Shen, Z. Li, J. Liu, and J. Grant, "Knowledge Sharing in the Online Social Network of Yahoo! Answers and Its Implications," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1715–1728, Jun. 2015.

[2] D. Movshovitz-Attias, Y. Movshovitz-Attias, P. Steenkiste, and C. Faloutsos, "Analysis of the reputation system and user contributions on a question answering website: Stackoverflow," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 886–893.

[3] Z. Li, H. Shen, and J. E. Grant, "Collective intelligence in the online social network of yahoo! answers and its implications," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 455–464.

[4] A. Bosu, C. Corley, D. Heaton, D. Chatterji, J. Carver, and N. Kraft, "Building reputation in StackOverflow: An empirical investigation," in *2013 10th IEEE Working Conference on Mining Software Repositories (MSR)*, May 2013, pp. 89–92.

[5] D. Cheng, M. Schiff, and W. Wu, "Eliciting Answers on StackOverflow," 2013.

[6] J. Yang, K. Tao, A. Bozzon, and G.-J. Houben, "Sparrows and owls: Characterisation of expert behaviour in stackoverflow," in *User Modeling, Adaptation, and Personalization*. Springer, 2014, pp. 266–277.

[7] R. K. Saha, A. K. Saha, and D. E. Perry, "Toward understanding the causes of unanswered questions in software information sites: a case study of stack overflow," in *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. ACM, 2013, pp. 663–666.

[8] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2011, pp. 2857–2866.

[9] P. Jurczyk and E. Agichtein, "Discovering authorities in question answer communities by using link analysis," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 919–922.

[10] H. Zhu, E. Chen, H. Xiong, H. Cao, and J. Tian, "Ranking user authority with relevant knowledge categories for expert finding," *World Wide Web*, vol. 17, no. 5, pp. 1081–1107, 2014.

[11] G. Zhou, S. Lai, K. Liu, and J. Zhao, "Topic-sensitive probabilistic model for expert finding in question answer communities," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 1662–1666.

[12] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen, "Probabilistic question recommendation for question answering communities," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 1229–1230.

[13] J. Guo, S. Xu, S. Bao, and Y. Yu, "Tapping on the potential of q&a community by recommending answer providers," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 921–930.

[14] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 261–270.

[15] F. Xu, Z. Ji, and B. Wang, "Dual role model for question recommendation in community question answering," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 771–780.

[16] L. Yang, M. Qiu, Swapna Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen, "Cqarank: jointly model topics and expertise in community question answering," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 99–108.

[17] T. C. Zhou, M. R. Lyu, and I. King, "A classification-based approach to question routing in community question answering," in *Proceedings of the 21st international conference companion on World Wide Web*. ACM, 2012, pp. 783–790.

[18] B. Li and I. King, "Routing questions to appropriate answerers in community question answering services," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1585–1588.

[19] S. Chang and A. Pal, "Routing questions for collaborative answering in community question answering," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013, pp. 494–501.

[20] A. El-Korany, "Integrated expert recommendation model for online communities," *arXiv preprint arXiv:1311.3394*, 2013.

[21] D. van Dijk, M. Tsagkias, and M. de Rijke, "Early detection of topical expertise in community question answering," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 995–998.

[22] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios, "Finding expert users in community question answering," in *Proceedings of the 21st international conference companion on World Wide Web*. ACM, 2012, pp. 791–798.

[23] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[24] M. Bouguessa, B. Dumoulin, and S. Wang, "Identifying authoritative actors in question-answering forums: the case of yahoo! answers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 866–874.

[25] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 221–230.

[26] D. W. McDonald and M. S. Ackerman, "Expertise recommender: a flexible recommendation system and architecture," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 2000, pp. 231–240. [Online]. Available: http://dl.acm.org/citation.cfm?id=358994

[27] C.-Y. Lin, N. Cao, S. X. Liu, S. Papadimitriou, J. Sun, and X. Yan, "Smallblue: Social network analysis for expertise search and collective intelligence," in *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*. IEEE, 2009, pp. 1483–1486.

[28] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[29] N. Novielli, F. Calefato, and F. Lanubile, "Towards discovering the role of emotions in stack overflow," in *Proceedings of the 6th International Workshop on Social Software Engineering*. ACM, 2014, pp. 33–36.

[30] K. Bajaj, K. Pattabiraman, and A. Mesbah, "Mining questions asked by web developers," in *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 112–121.

[31] A. Joorabchi, M. English, and A. E. Mahdi, "Automatic mapping of user tags to Wikipedia concepts: The case of a Q&A websiteStackOverflow," *Journal of Information Science*, p. 0165551515586669, 2015.

[32] D. Posnett, E. Warburg, P. Devanbu, and V. Filkov, "Mining stack exchange: Expertise is evident from initial contributions," in *Social Informatics (SocialInformatics), 2012 International Conference on*. IEEE, 2012, pp. 199–204.

[33] Z. Zhao, L. Zhang, X. He, and W. Ng, "Expert Finding for Question Answering via Graph Regularized Matrix Completion," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 27, no. 4, pp. 993–1004, 2015.

[34] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the web." 1999.

[35] W.-C. Kao, D.-R. Liu, and S.-W. Wang, "Expert finding in question-answering websites: a novel hybrid approach," in *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 2010, pp. 867–871.

[36] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[38] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[39] X. Liu, W. B. Croft, and M. Koll, "Finding experts in community-based question-answering services," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 315–316.

[40] T. Zhao, N. Bian, C. Li, and M. Li, "Topic-Level Expert Modeling in Community Question Answering." in *SDM*, vol. 13. SIAM, 2013, pp. 776–784.

[41] M. Thorup and U. Zwick, "Approximate Distance Oracles," *Proceedings of the ... Annual ACM Symposium on Theory of Computing.*, vol. 33, pp. 183–192, 2001, oCLC: 108197731.

[42] D. Krioukov, K. Fall, and X. Yang, "Compact Routing on

Internet-Like Graphs," *Proceedings /*, vol. 1, pp. 209–219, 2004, oCLC: 108329169.

[43] Dmitri Krioukov, k claffy, Kevin Fall, and Arthur Brady, "On compact routing for the internet," *SIGCOMM Comput. Commun. Rev. ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 41–52, 2007, oCLC: 4761136181.

[44] P. Fraigniaud, E. Febhar, and F. Viennot, "The inframetric model for the internet," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE.* IEEE, 2008.

[45] P. Assouad, "Plongements lipschitziens dans $bbfR^n$," *Bulletin de la Société Mathématique de France*, vol. 111, pp. 429–448, 1983.

[46] A. Slivkins, "Distance estimation and object location via rings of neighbors," *Distributed computing.*, vol. 19, no. 4, p. 313, 2006, oCLC: 103428197.

[47] D. Xia, "Compact routing design in networks of low doubling dimension," Ph.D. dissertation, 2008, oCLC: 740506433.

[48] I. Abraham, C. Gavoille, A. Goldberg, and D. Malkhi, "Routing in Networks with Low Doubling Dimension," *26th IEEE International Conference on Distributed Computing Systems*, p. 75, 2006, oCLC: 4799415157.

[49] G. Konjevod, A. W. Richa, and D. Xia, "Optimal-Stretch Name-Independent Compact Routing in Doubling Metrics," *PROCEEDINGS OF THE ANNUAL ACM SYMPOSIUM ON PRINCIPLES OF DISTRIBUTED COMPUTING*, no. Conf 25, pp. 198–207, 2006, oCLC: 203087469.

[50] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[51] S. T. Dumais, "Latent semantic analysis," *ARIS Annual Review of Information Science and Technology*, vol. 38, no. 1, pp. 188–230, 2004, oCLC: 5154921734.

[52] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The Measurement of Textual Coherence with Latent Semantic Analysis." *Discourse Processes*, vol. 25, no. 2-3, pp. 285–307, 1998, oCLC: 425240089.

[53] P. Jacquet, "Common words between two random strings," Ph.D. dissertation, INRIA, 2006.

[54] G. Burnside, D. Milioris, P. Jacquet, and others, "One Day in Twitter: Topic Detection Via Joint Complexity." in *SNOW-DC@ WWW*, 2014, pp. 41–48.

[55] J. Darlington, "A synthesis of several sorting algorithms," *Acta Informatica*, vol. 11, no. 1, pp. 1–30, 1978.

Appendices

1. The Travel StackExchange dataset

Before performing all the analyses with the main dataset for this work, we use a smaller set from the same family as an exploration laboratory, i.e. the Travel StackExchange dataset from December 2015 (https://archive.org/details/stackexchange). It contains 12736 questions, 22655 answers, 8233 users and 71255 comments. In this appendix we present the results obtained when applying the same processes as in the main work in order to contrast both datasets, since interesting qualitative differences seemed to be embedded in the data.

1.1. Cover number

The most striking difference comes from the first result, dealing with the possibility of having a bounded cover number. In Figure 1 we observe a a much clearer correlation between the size of L_u and the cover number. This plot seems to indicate that for this dataset the cover number is not bounded. Smaller reduced set of interests might explain this result. If most users only comment and ask questions, it is harder for users with diverse interests to find groups of other users who can answer their questions.

The difference points to a qualitative mismatch between both datasets. The community of Math StackExchange seems to be more engaged, with abundant users willing to answer questions, whereas it seems to be the case that people go into Travel Exchange just for the answers. This can be confirmed by looking at Figure 4, since the difference in size for the large set and the reduce set of common interests for the users is significantly larger than in the Math case. It can be argued that a healthier environment for obtaining answers is found when the cover number is bounded, which at the same time correspond to better conditions for routine schemes.

1.2. User's history

Now we explore the results dealing with user histories and their sets of interests. The number of answerers is even smaller, as is shown in Figure 2, which is not surprising for a smaller and less technical website. For the same reason poorer results are to be expected when looking for answerers in the reduced and large sets of common interests. This intuition is corroborated in Figure 3. It seems to be a harder task to find answerers in this website.

There are no significant difference for the distribution of user load. It is again shown that overload is mostly avoided, with only a handful of users being responders of a significant number of questions, as presented in Figure 5.

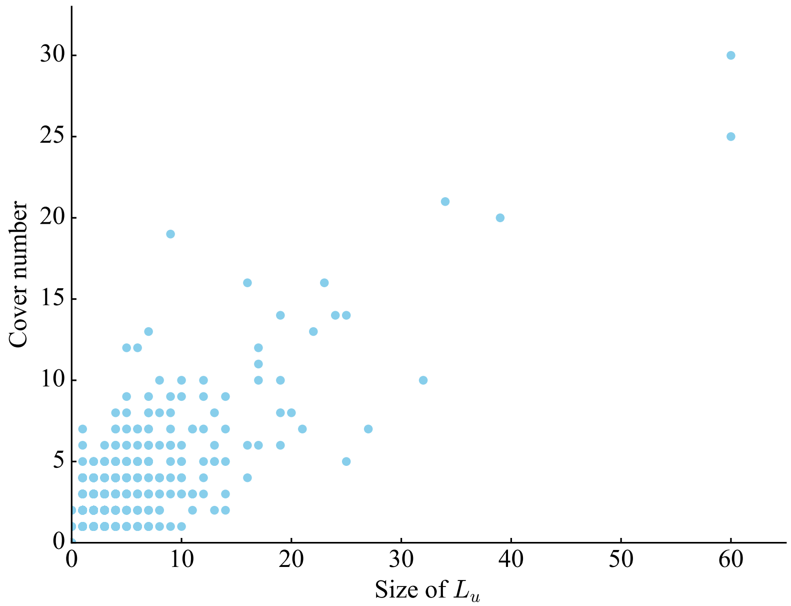

FIGURE 1: Cover number vs $|L_u|$ for all $u \in U$. This number presents a clear incremental tendency, thus a bounded cover number is not to be expected in this dataset.

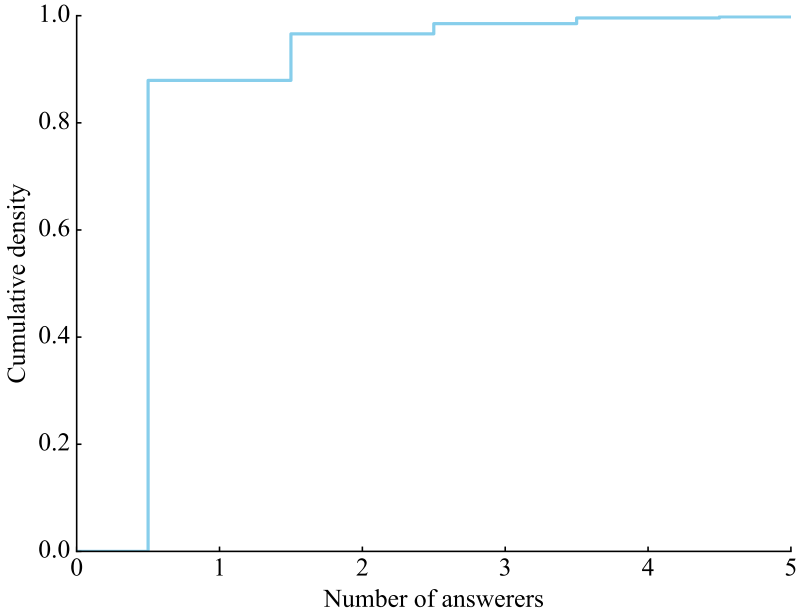

FIGURE 2: Cumulative density of the number of responders per question. It is seen that above 90% of the questions have less than 3 answerers.

1.3. Similar questions

The results when looking for similar questions through the order given by Joint Complexity distances, presented in Figure 6, are similar, although the effect of the size of the dataset is noticeable. More answerers are found within the first 50 questions, approximately 60%. Nevertheless, selecting questions at random performs much better than in the original dataset, which can be understood as a size effect.

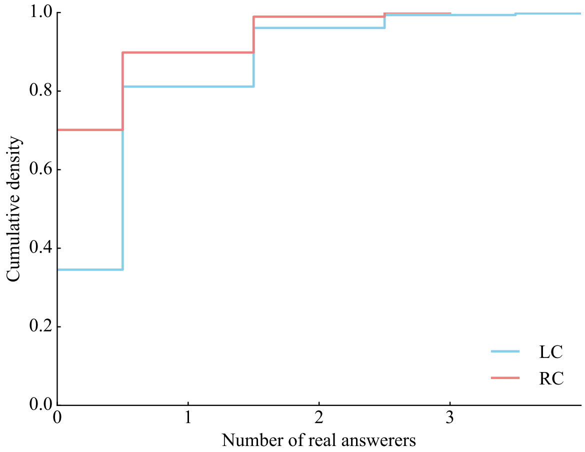

FIGURE 3: Cumulative density of the number of real responders found in LC_u and RC_u. By exploring LC it is possible to find a responder for bit over 60% of the questions. This result is not near as good at the one obtained for Math StackExchange

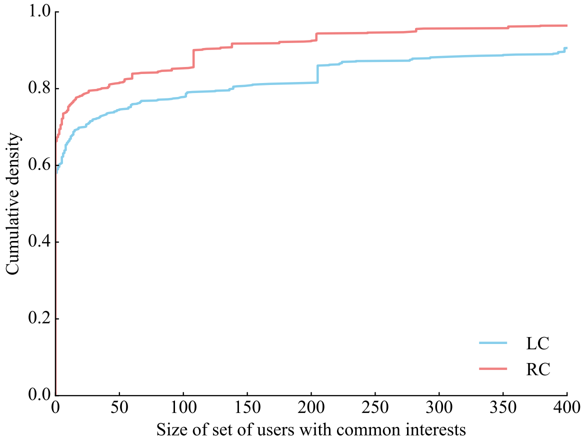

FIGURE 4: Cumulative density of $|LC_u|$ and $|RC_u|$. The tradeoff between recall and the number of potential candidates to consider is illustrated. Considering LC obliges to browse about three times the number of users.

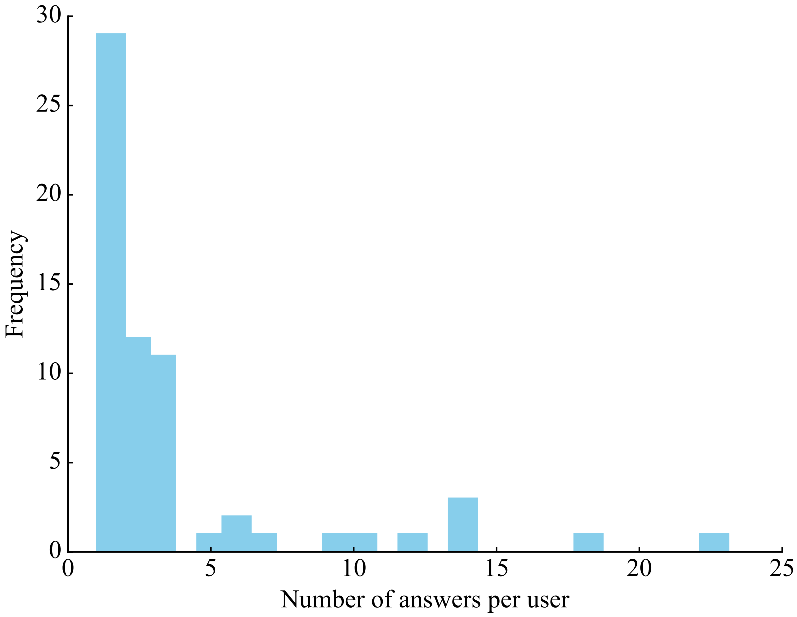

FIGURE 5: Histogram of the number of answers given per user. Although there exists a small number of highly requested users, there is a considerable number of users providing answers.

Regarding the analysis of question load on users, we observe in Figure 7 that the results are very similar to both the main dataset and the distribution of load for the graph approach. No significant overload is observed and the presence of many interacting users is confirmed.

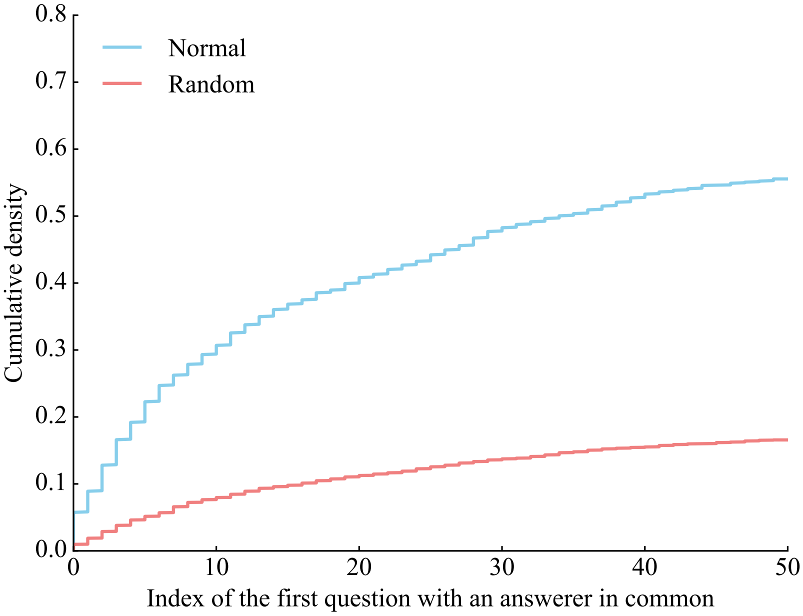

FIGURE 6: Cumulative density of the index of the first question with a responder in common with a given question. The performance of the algorithm that order the questions according to distance is considerably better than when a random selection is taken.

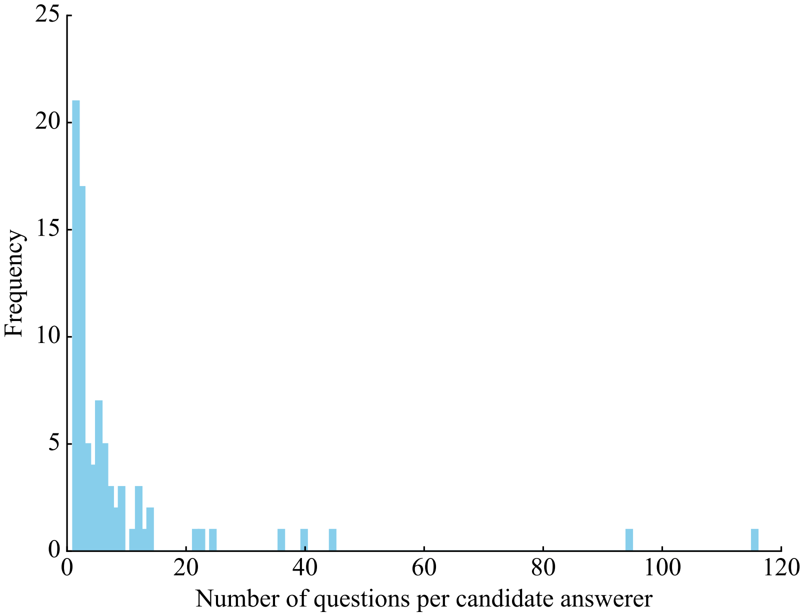

FIGURE 7: Histogram of the number of answers given per user. The same general features as in the graph approach are observed, in particular the avoidance of user overloading. Similar as well, although for a smaller set, than the results found in Math

1.4. Question routing algorithms

Figure 8 shows the results obtained when applying the algorithms to the travel dataset. As for the method of finding neighboring questions, there is better recall for all of the choices, including when selecting questions at random. This indicates again a size effect. Apart from this difference, the qualitative behavior of the algorithms is the same as the one found for Math StackExchange.

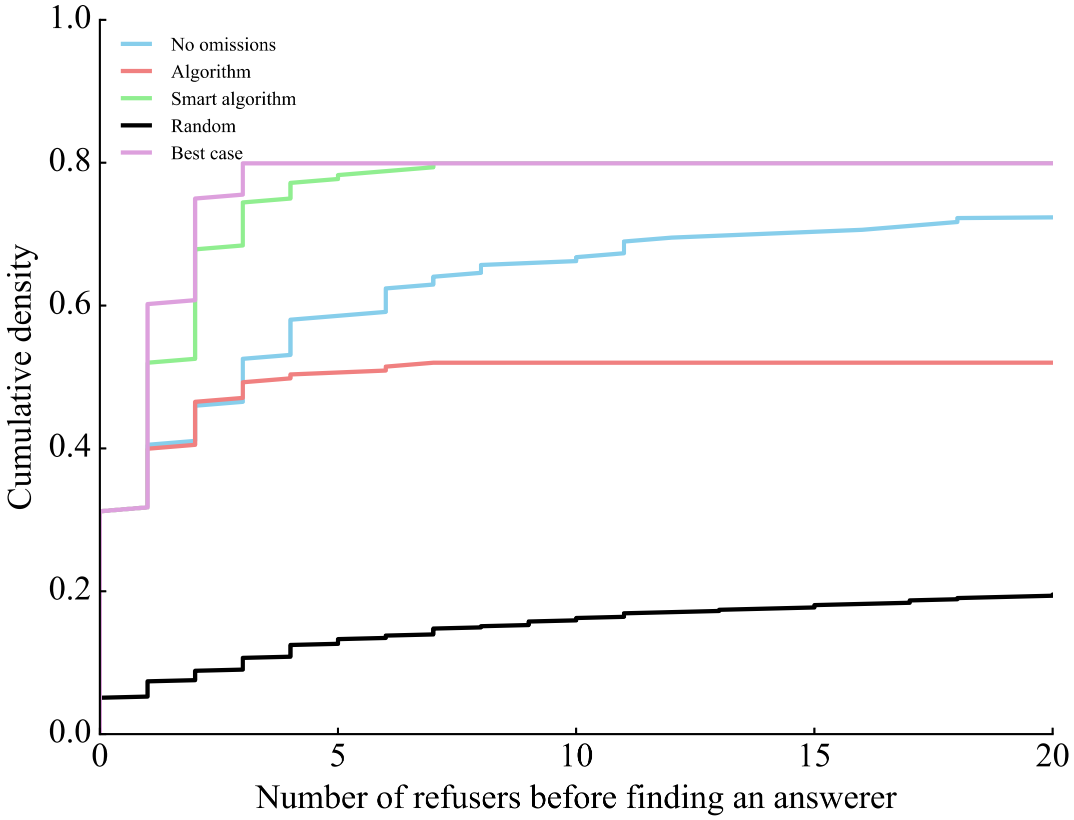

FIGURE 8: Cumulative histogram of users queried before finding a responder under different methods. The results are quite similar to the ones found for Math StackExchange, except for data size effects.

1.5. Conclusions

Although initially used as a data playground, the analysis of the travel dataset brought interesting surprises to the table. The geometry of the underlying space seems to be fundamentally different from the Math's space. This is possibly due to a difference in user behavior, with Math's users being more proactive in answering questions, and thus growing their respective reduced set of common interests. Therefore, it is desirable to have rich interactions within the system if a routing algorithm is being considered.

It is remarkable that this potential difference does not considerably affect the performance of the different algorithms that were proposed. The only important disagreements between the results are likely to be caused by data size effects. It is our believe that this results were interesting enough to make the addition of this data set worth the reader's attention.

2. Text metrics

In order to perform tasks such as text clustering it is common to use the Vector Space Model (Salton et al. 1975), which idea is to represent documents as vectors, so that relations between them can be established by calculating distances or performing projections. Arguably the most classical and still widely used method to achieve this embedding consists on building a so called TF-IDF (Term frequency-inverse document frequency) matrix. Its rows correspond to words while columns represent documents. A single value (t, d) in the matrix corresponds to the frequency of the term t in the document d weighted by its inverse frequency across the whole corpus. Thus, common words will have low values, while rare words that are common in only one document will have a high value in such place.

More formally, for a term t and a document d, we define tf $(t, d) = f_{t,d}$, where $f_{t,d}$ corresponds to the frequency of t in d. We further define idf $(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$, with D being the set of all documents and $N = |D|$. Finally we define tfidf $(t, d, D) =$tf$(t, d)\cdot$idf(t, D).

Once the matrix is built each document is represented by a vector, with the number of components corresponding to the total number of words in the vocabulary. Since this might be too extensive, it is possible to apply dimensionality reduction techniques such as using the Singular Value Decomposition of the matrix (Golub & Van Loan n.d.) and discard the columns corresponding to negligible eigenvalues.

After all this preprocessing it is simple to calculate distances or similarity between vectors, e.g. by using the cosine similarity between two vectors $s = \frac{\mathbf{x}\cdot\mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$. This approach capture some of the latent semantics of texts (Dumais 2004), which makes it a very powerful tool in problems such as information retrieval. However as the number of documents grows the matrix scales as the product of the number of documents and the number of unique words, which makes its scalability a challenge.

Hence, for the task of finding similar questions, and taking into account all the distances required, we opted to use a metric which does not involve information of the whole corpora. Joint Complexity is based on the complexity measure, which in turn identifies the amount of information in a sequence, without any regard for the meaning (Jacquet 2006). In fact, similar information retrieval metrics are used in genome sequencing (Allison et al. 2000) where no semantic is present. By measuring the amount of share information Joint Complexity offers an efficient way to identify neighboring texts.

There exist many other text metrics where to choose from depending on the task on hand. Moreover, depending on the features we handle in certain problems, additional considerations can be added tot he metric, e.g. allocate weights depending on whether the answerers of a question visit the site frequently, which would result in the preference for questions that have been answered by highly active users, who are in turn more likely to answer the new inquiry. As a final remark, Joint Complexity proved to be sufficiently good to illustrate the potential of the proposed framework, even when only considering the titles of the questions, which further confirms that it is a good tool to use in tasks where short texts are present.

3. The role of topics

3.1. Introduction

A significant part of the present project was dedicated to exploring the potential of topic detection algorithms for question routing. The general idea was to identify whether the problem could be split into topics, i.e. use the topic of the incoming question in order to select users whose interest lies in the respective topic as potential responders. We wanted to explore a different approach than that of state of the art probabilistic topic models (Blei 2012), where topics are identified as distributions over words and texts are in turn represented as distributions over topics. In this case topics are latent, with texts being related with multiple topics at the same time.

The research in these kind of models has been extensive with highly specific and efficient methods being developed in recent years. Hence, our idea was to look for more general and complementary solutions, which could integrate such models, instead of competing directly with them in a certain website arquitecture.

Thus, we explored a more direct approach, with the idea of implementing simple algorithms which could scale better than the processes required to learn such models. The first task consisted on using standard clustering techniques in order to see if posts could be clustered in a logical fashion according to their main topic. To begin with the texts most be parsed, tokenized and filtered so as to eliminate mistakes, code or stop words among others. This preprocessing was performed based on a tutorial provided by Brandon Rose (http://brandonrose.org/clustering).

Afterwards text metrics have to be considered in order to either embed the texts in a metric space or to find pairwise similarities between them. Several algorithms were used in order to guarantee consistency and understand the clustering properties of the data, thus both possibilities were considered, with the choice made according with the specific requirements of each algorithm. When a method required a metric space, we opted for the classic approach in the Vector Space Model explained in the former section. Otherwise we used Joint Complexity to define a similarity.

3.2. First approach: K-means

The first algorithm to be considered was arguably the most popular one, i.e. K-means (MacQueen et al. 1967), which despite its simplicity keeps being used in several studies due to its efficiency and scalability. We used the specific implementation found in the Python library Scikit-learn, which performs some additional processing to stabilize the final clusters under the random choice of initial centroids. This algorithm requires as an input the number of clusters. A direct trial and error method and silhouette analysis (Rousseeuw 1987) were combined to decide on a reasonable number of clusters.

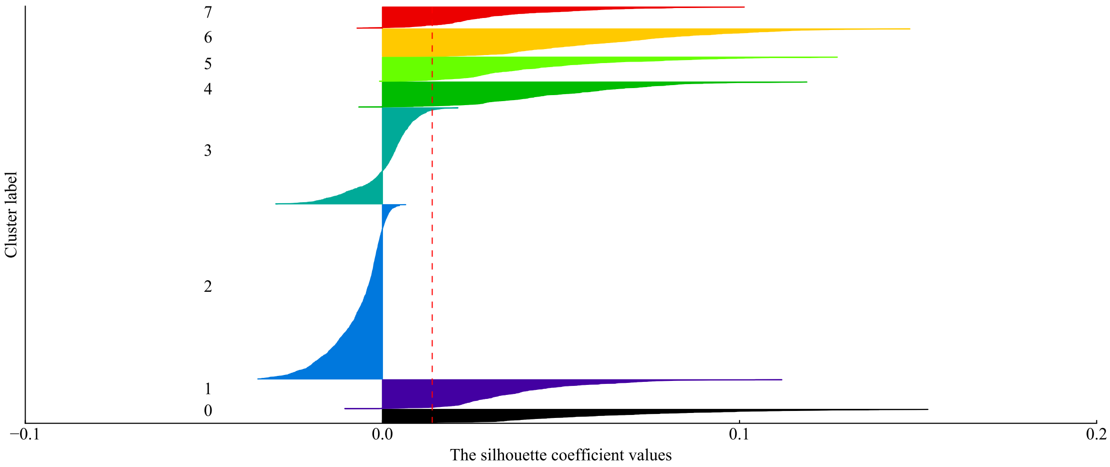

FIGURE 9: Silhouette analysis for k-means with $k = 8$. It is observed that there are two clusters which are considerably bigger than the rest and whose silhouettes imply a problem with this particular number of clusters.

Figure 9 show an example of the output of this algorithm, which illustrates a persistent problem with this clustering approach: notwithstanding the number of clusters selected, from 4 to 100, there were always one or two clusters significantly bigger than the others. These results are related with the findings in (Barua et al. 2014) where two of the obtained clusters where 'topicless', in that there was no common topic between the popular words identifying them. To identify the topic of a cluster we printed the five words with highest frequency in the vector representing the centroid. For example, one such cluster consisted of the words *visa, embassy, border, immigration, passport*, which would clearly identify the cluster as dealing with immigration issues for international travel. As in the cited study, our big clusters did not have a discernible topic, with their most frequent words being very generic, e.g. *good, travel, happy, water, true.*

The problem in our particular case is that these clusters tended to encompass up to 30% of the texts, thus generating a big data loss. Although all the other clusters were identifiable with logical topics in the travel theme, we decided to use other algorithms in an attempt to solve this issue. Furthermore, the same results were obtained for subsets of the Math StackExchange dataset, indicating that it was not an artifact of the particular texts used in the travel forum.

3.3. Other algorithms

In order to find some answers related to our big general clusters, we applied a plethora of clustering algorithms:

- Hierarchical clustering (Johnson 1967): due to the possibility of visualization that this algorithm provides, the results were quite informative in order to understand the problem. The big clusters contained smaller clusters of sizes closer to the others. Nonetheless, in order to arrive at the level of those desired clusters, the resolution had to be reduced enough for the other clusters to subdivide in even smaller ones. This in turn explains the persistency of this issue at all scales.

- Spectral clustering (Ng et al. 2002): this graph based approach yielded a recognizable improvement, however it did not entirely solve the problem and using this algorithm would mean sacrificing scalability. Both the VSM and JC were used with similar results.

- Louvain method (Blondel et al. 2008): this is a community detection algorithm for networks. It was implemented in the graph of questions, where two question are linked if the same user participated in both. Although an efficient method, the communities detected presented even more cases of lack of topics.

- Latent Dirichlet Allocation (Blei et al. 2003): this is not a clustering algorithm but the quintessential probabilistic topic model. The goal of implementing it was to verify that the topics detected by other methods corresponded to the ones found via this generative model. For K-means and spectral clustering the correspondence was almost one to one when the number of clusters was selected as the number of topics provided by this method.

- Ensemble methods (Vega-Pons & Ruiz-Shulcloper 2011): ensemble methods combine the results of a clustering algorithm performed multiple times with different initial condition in order to find robust clusters. We implemented this algorithm for ensembles of K-means and spectral clustering, but no improvement was obtained.

3.4. Lack of cluster correspondence

In order to delve further into the role of topics, we decided to choose a relatively big number of clusters, in the order of 40, to minimize the size of the 'topicless' clusters without loosing the coherence in topics for the rest of the clusters. The next question to answer was: do clusters of questions correspond to clusters of answers? For topic identification to be useful we do need that the topic detected for a question be the same as the one detected for its corresponding answers.

To answer this question we clustered independently the set of questions and the set of answers (using both k-means and spectral clustering) and manually identified the corresponding clusters, e.g. the cluster of questions identified by *visa, border, passport* would be deemed to corresponds with the cluster of answers represented by *visa, embassy, control*. Ideally all the answers for a given question would be in its corresponding cluster. Of course, in real datasets the ideal case is hardly, if ever, the result. Nevertheless, one could expect a correspondence of over 70%, which would imply that the approach still has some potential.

However, the results were quite far from the expected behavior. The maximum correspondence obtained for a pair of identified clusters was 55%, i.e. only 55% of the questions to the corresponding answers were in the correct cluster. Even more striking, the average correspondence was between 20% and 25%, which renders methods for using this hard cluster topics quite likely to be unsuccessful. On the bright side, for almost all of the clusters in all of the cases analyzed, the biggest correspondence with other cluster was obtained precisely for the appropriate ones.

3.5. Conclusions

Although sensible topics were identified via clustering, the facts that a big number of posts get clustered in 'topic less' groups together with the lack of correspondence between clusters of questions and clusters of answers discourage us to follow this path in look for potential methods of question routing. Instead it is our believe that topic information can be used as part of a model such as is successfully done in existing topic models, or as a weight factor in text metrics used to identify neighboring posts. In this latter case it might be useful to trust in the good sense of the communities in the case where they provide tags, in order to avoid or improve the automatic cluster detection.

References

Allison, L., Stern, L., Edgoose, T. & Dix, T. I. (2000), 'Sequence complexity for biological sequence analysis', *Computers & Chemistry* **24**(1), 43–55.

Barua, A., Thomas, S. W. & Hassan, A. E. (2014), 'What are developers talking about? an analysis of topics and trends in stack overflow', *Empirical Software Engineering* **19**(3), 619–654.

Blei, D. M. (2012), 'Probabilistic topic models', *Communications of the ACM* **55**(4), 77–84.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *the Journal of machine Learning research* **3**, 993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. (2008), 'Fast unfolding of communities in large networks', *Journal of statistical mechanics: theory and experiment* **2008**(10), P10008.

Dumais, S. T. (2004), 'Latent semantic analysis', *ARIS Annual Review of Information Science and Technology* **38**(1), 188–230. OCLC: 5154921734.

Golub, G. H. & Van Loan, C. F. (n.d.), ‘Matrix computations. 1989’.

Jacquet, P. (2006), Common words between two random strings, PhD thesis, INRIA.

Johnson, S. C. (1967), ‘Hierarchical clustering schemes’, *Psychometrika* **32**(3), 241–254.

MacQueen, J. et al. (1967), Some methods for classification and analysis of multivariate observations, *in* ‘Proceedings of the fifth Berkeley symposium on mathematical statistics and probability’, Vol. 1, Oakland, CA, USA., pp. 281–297.

Ng, A. Y., Jordan, M. I., Weiss, Y. et al. (2002), ‘On spectral clustering: Analysis and an algorithm’, *Advances in neural information processing systems* **2**, 849–856.

Rousseeuw, P. J. (1987), ‘Silhouettes: a graphical aid to the interpretation and validation of cluster analysis’, *Journal of computational and applied mathematics* **20**, 53–65.

Salton, G., Wong, A. & Yang, C.-S. (1975), ‘A vector space model for automatic indexing’, *Communications of the ACM* **18**(11), 613–620.

Vega-Pons, S. & Ruiz-Shulcloper, J. (2011), ‘A survey of clustering ensemble algorithms’, *International Journal of Pattern Recognition and Artificial Intelligence* **25**(03), 337–372.