Dialog Markets

Andreas Stuhlmüller
Stanford University

September 2016

Abstract

People frequently ask Google questions like "What career is right for me?" "How can I find a boyfriend?" or "How can I make more money?" While Google can answer self-contained questions like "What is the capital of Turkey?" it cannot answer vague questions, even though these tend to be the questions people care about the most. Answering these questions requires background information about the person who asked. Good advice is tailored to the particulars of the asker's situation.

A natural way to elicit such background is through a dialog in which follow-up questions are asked. If you ask a doctor "Why do I have stomach pain?" they won't just give you an answer. They will ask clarifying questions: "When did it start?" — "Where is it?" — "Is it constant or intermittent?" And only then they suggest possible causes.

In this set of notes, I propose *dialog markets*, a mechanism for creating high-quality conversations that resolve vague questions.

A dialog starts when an individual asks a question and pledges a reward that will be used to pay for helpful contributions. Humans and machines then contribute follow-up questions and other responses. The core problem that the market mechanism needs to solve is how to distribute reward over contributions such that we incentivize conversations that are as valuable as possible for the asker. We can view this as a particular set of questions that we can delegate to the market: "How much should the asker pay for this contribution?" This view gives rise to the grounding problem for dialog markets: incentives will be aligned if answers to this meta-level question are good, but that in turn depends on aligned incentives. I outline some approaches towards addressing this problem.

Finally, I suggest an implementation plan: I discuss how to ensure that early dialogs have sufficiently high quality to get the project off the ground and name a few potential initial applications.

Contents

1 Introduction

1.1 The problem: answering vague questions

There are many questions we ask in the course of our lives. For example, there are personal questions, big and small: *Which college should I go to? What kind of career is right for me? What should I do in my free time? How can I find a significant other? What mobile phone should I buy? What movie should I watch next? Why do I have stomach pain? Who should I vote for?* There are questions that come up in the course of our work: *How can I get better at writing? What programming language should I use? What should I work on next?* There are big questions about humanity and its place: *How did the universe come to be? What are the key scientific and technological developments of our time? When will AI surpass human abilities in most fields? Are humans causing global warming?* And there are questions we idly wonder about: *Which team is going to win the world cup? What is the best game for the SNES?*

Most of these questions are vague: that is, there is relevant information about what the question means that is not explicitly expressed. This doesn't mean that some ways of answering these questions aren't much better than others—it just means that the best ways of answering will take into account context.

1.2 Dialogs clarify questions

Most of the questions mentioned above are difficult to answer on their own. If you ask a doctor *"Why do I have stomach pain?"*, they won't just give you an answer. They will ask clarifying questions: When did it start? Where is it? Is it constant or intermittent? Is it getting worse or better? Tell me what your pain feels like. Is it dull or sharp? Is it localized or diffuse? How severe is it? Is there anything that makes it better, or worse? Do you have any other symptoms? Weight loss? Vomiting? Fever or chills? And then they suggest possible causes.

So, an important part of answering questions is figuring out what exactly the question means, and what additional information there is that could help answer the question. Typically, this happens through a dialog in which clarifying questions are asked. In this project, we will think about how to set up a system that produces dialogs that lead to high-quality answers. Before we make plans for such a system, let's first think about another important role that dialogs can play. If you're in a rush, you may want to jump ahead to Section 3, which gives a taste of the overall proposal.

1.3 Dialogs prompt relevant mental computations

When someone asks you a question, you think for a bit, then answer. Note how little time passes between question and answer, in particular for real-time dialog. Only very few steps of mental computation can take place between question and answer; and yet, you can answer complex questions like "Do you want to go on a date with me?" in a second or two. Think about all the considerations that would go into answering a question like this if you had unlimited time. What will my future look like if I say yes? What if I say no?

And yet, you can produce an answer for every question in the allotted time, even if it is just a "I don't know". Most of the computations used must already have taken place, in your head, in someone else's, or over the course of human evolution, and you must have cached some of them for future reuse. When we talk about "intuition", maybe we're talking about what such cached computations feel like from the inside.

This reuse of cached computations is likely approximate. In Thinking, Fast and Slow, Kahneman discusses such approximations: "If a satisfactory answer to a hard question is not found quickly, System 1 will find a related question that is easier and will answer it." To give a few examples: "What are the best careers for making a lot of money?" becomes "What careers have I come to associate with wealth?"; "How happy are you with your life these days?" becomes "What is my mood right now?"; and "How frequently does this happen?" becomes "How easily do instances of this come to mind?" (via Kaj Sotala).

Within a conversation, time and resource constraints are similar for questions that require few computational steps to answer and questions requiring many steps. We should expect that, all else being equal, the former will in general get more accurate answers unless pretty much all of the necessary computational steps for the big questions have been cached at some point in the past. More commonly, there will be approximate reuse of computation for at least some steps, so we should expect higher error for big questions.

The same reasoning applies to actions: When we make decisions in real-time, very few uncached computation steps are feasible. For decision quality to be high, we therefore need good cached computations. The source of such computations can be previous computation online (when we made similar decisions), thinking offline (reflection), and external information sources (e.g., things we read on the Internet, or discussions with friends). Consider emergency doctors and martial arts practitioners: almost instantly, they make decisions that you would have no chance of getting right, and what allows them to do this is that they invested years into building the right kinds of cached computations.

Let's summarize: questions induce computations, and computations partially reuse stored computations. This reuse is approximate when no exact match can be found. If we are interested in the answer to a big question—"What do you want to do with your life?", say—then directly asking may not lead to good answers; a good answer would take a lot of computation, and there is generally little time available to produce an answer, so if the person being asked has not performed the necessary subcomputations previously, they will have to produce an answer using their best-matching stored subcomputations, which can lead to low-quality answers.

By asking the right kinds of questions in a dialog, we might be able to induce computations ahead of time that lead to better answers and decisions when it counts.

2 Strategies for Question-Answering

Our goal is to design a system that can answer vague questions. The considerations in the previous section suggest that dialogs may be an effective tool for this purpose. This doesn't constrain the space of possible systems enough for practical work. So, let's think about what strategies an effective system for question-answering could employ.

2.1 Break down big questions into small ones.

Based on the argument in Section 1.3, we should expect that decomposition results in better answers when the question cannot be answered correctly based on mostly stored computations. For example, suppose you are thinking about some negative state of things x. "What could you do to make it less likely that x?" is a big question. If you haven't thought much about the causes and components of x, and if you haven't received high-quality external information on how to reduce the probability of x, then it will be difficult for you to produce a good answer. This could be easier if we first ask about the causes and components of x.

2.2 Provide answers given by others to turn production tasks into judgment tasks.

In various circumstances, it is easier to recognize good answers than to produce them. For three very different examples, see (1) Ira Glass on The Gap between having taste and producing work that is good according to one's taste, (2) Justice Potter Stewart's famous "I know it when I see it", and (3) the strong belief that P!=NP in computational complexity.

In circumstances like this, providing answers that other people have given to a question can make it much easier for you to answer the same question. In cases where it is difficult to break down a complex question, it may still be the case that some people manage to answer the complex question, and that others can recognize a plausibly good answer when they see it, even if they couldn't have produced it themselves.

There is a virtuous interaction between decomposing questions and sharing answers: the more we decompose a question, the more we should be able to reuse answers to subquestions.

2.3 Externalize thoughts to remove memory constraints

Imagine that you are planning how to drive to a new place at the other end of the city without a map. Even if you know all the roads, it will be difficult to find the best route, just because it is difficult to keep all the information in your working memory at once. You can think about all of the individual roads, one by one, but you will probably not choose an ideal route because you can't quite keep in mind their connections.

Similarly, when you think about your own thoughts and behaviors, you may be able to analyze any given thought or behavior, but locally optimizing one of them may lead to problems with others. It may be easier to see connections, and to prioritize across possible thoughts, if thoughts are represented externally. Free-form writing (as in diaries) and mind maps are two existing tools that serve this function.

2.4 Externalize thoughts to enable incremental progress within individuals

Thoughts are fleeting. When we don't write down our thoughts, our recollection will frequently be spotty. This makes progress difficult: whenever we think about some topic, we have to first reconstruct our previous state of investigation. If our recollection is sufficiently lossy, we may not make any progress at all; recreating our previous thoughts may take too long for us to bother. By writing thoughts down, or otherwise giving them more persistent form, we make this process easier, and we can more easily make incremental progress.

2.5 Share thoughts to enable incremental progress across individuals

Externalizing thoughts also enables incremental progress across people. Compare how we think about our personal lives to how science works: in science, there is a continuously growing global store of knowledge that individual scientists build on and contribute to. It would be a terrible idea for a scientist to discard all of the knowledge that their field has acquired, or to only acquire this knowledge from pop science articles and very occasional conversations with other scientists.

On the other hand, this is the default state for our personal lives: we read advice here and there, and we have occasional conversations with friends and family, but in the end, our choices are quite independent. Each of us contributes very little to humanity's knowledge of how to live good lives, and even those who try don't systematically build on what others have discovered before.

We do have a few systems that go part of the way: Online communities such as Reddit, Quora, and various forums discuss personal matters and allow in principle for progress to be made, but they suffer from a combination of low intellectual standards, fragmented discussions, misaligned incentives (posts should be funny, surprising, impressive), and limited access to personal details. Parts of the quantified-self movement have higher intellectual standards, and the movement is more personalized by nature, but it limits itself to what can easily be measured. The field of positive psychology, and possibly some non-academic publications on self-help, have arguably made some progress over the years, but much of the thinking is on a very coarse level ("Does marriage make people happier?") and there is a clear separation between naive consumers and expert producers, which limits the number of producers.

2.6 Communicate thoughts to make them more coherent

Putting thoughts in words has benefits beyond resolving memory constraints and sharing acquired knowledge: when I think about what words will reconstruct an idea in the reader's mind, I often find that I do not understand the idea very well myself.

I don't understand this process very well—why does writing down ideas for the purpose of communication cause me to notice their weaknesses? Here are two guesses: First, the external representation needs to be sufficient to reconstruct the idea when combined with the reader's background knowledge, which forces me to look at all parts of the idea, even ones that I assumed to be clear without much explicit consideration. Second, social considerations create an adversarial setting: when I know that my ideas will be examined by others, I am more motivated to find their flaws so that I can fix them before they do.

2.7 Let machines do some of the work

At this point in time, we don't have good ways to represent and reason about arbitrary human knowledge in machines, so most reasoning will have to happen in human brains. However, we can strive to automate some relatively easy inferences, and extend this over time as the science of knowledge representation advances.

For example, if x is known to be good, and y is known to cause x, then we can infer (heuristically) that y instrumentally inherits some of the goodness of x, and we may not need to ask about the goodness of x explicitly.

More generally, whenever you notice that your thinking follows a pattern, consider automation. For example, you may notice that, when you are brainstorming, you ask a relatively fixed sequence of questions such as: (1) Write down anything that comes to mind. (2) Write down reasons why it is hard to come up with ideas. (3) Make an exhaustive taxonomy of things you could write. (4) Randomly constrain your problem (5) Etc. Outsourcing the question-asking process to machines can make it more reliable, and it makes it easier to share and incrementally improve the process itself.

Another place where machine learning can be applied is in learning which questions to ask: Record for each question whether people answer it, or whether they press "skip". An answer is interpreted as a positive signal, "skip" as a negative signal. Each question is associated with a number of features (e.g., words used, the question type, context signals). We can use supervised machine learning techniques such as logistic regression to preferentially ask questions that are most likely to get answered.

2.8 Focus attention on topics with high expected payoff

There are many topics one could reflect on, and for any given topic, there are many questions one could ask. Without systematic efforts, we have little reason to believe that naive attempts at reflection will pick the most promising topics, and little reason to believe that we will ask the best questions for any given topic. It is likely that both choices will instead be driven by what easily comes to mind. This availability heuristic can be problematic in cases where we would like to look at our lives with fresh eyes, and where we would like to prioritize based on where we can make the most progress. Indeed, some of the questions we should ask may be the questions we least want to ask, e.g. because they have inherited some of the aversiveness of their subject matter. If I hate doing my taxes, then at some point the very thought may become aversive enough that I flinch away from it every time my thoughts get close. We can probably do better by following a more systematic process for choosing topics and questions, in particular if we can learn from what other people found helpful.

In the remainder of this document, I will lay out a proposal that uses the techniques above and that is intended to produce high-quality answers for vague questions.

3 Three Fictional Perspectives

Before I go into the proposal per se, I will describe it from the perspective of three hypothetical users. This is intended as an introduction of the proposal that is easier to read at the cost of leaving out critical parts that are less visible to users.

3.1 A person with questions

"So I have a question I'm wondering about, really any kind of question. Recently, I've used it to better understand what's wrong with my stomach before going to the doctor, and to figure out where we should go for our vacation.

I only use their app on my phone (Figure 1). I think they have a web page as well. I type in my question. Sometimes it immediately asks follow-up questions, sometimes it directly proposes an answer, but most of the time, it takes a while. If I care a lot about the answer, I can pledge money and I get better answers and follow-up questions.

Every now and then, I get a push notification on my phone—there are new comments on my question, or new follow-up questions for me. Most follow-up questions are simple multiple-choice questions, so it's easy to answer them directly on my phone.

Over time, this builds up a dialog related to my initial question and I get more specific questions and proposals. Actually, I only participate in a small part of the process, the part that requires info about my life, but there are many subquestions that get resolved without me.

I don't know where exactly the questions and answers are coming from—I think some are generated by computers, and some by humans? I just know that the dialogs are pretty helpful, like talking to a knowledgeable friend, at least for questions that a lot of other people have, too.

For more unusual questions, the dialogs take more time; in general, it's not a great tool when I am in a hurry. But if I have time, it's neat—I can ask a question, pay \$10, get asked a bunch of easy-to-answer multiple-choice questions, and then get back a well-developed analysis of my question specific to my situation. Then, if I want to see more thought put into any subpart of the answer, I can just pay money towards the relevant subquestion and it will be used to reward mental work that helps solve it."

3.2 A domain expert with answers

"I'm a med student by day. I don't have much free time, but recently I've enjoyed participating in dialogs on that website (Figures 2, 3).

I tend to search for medicine-related questions that I might be able to answer, but sometimes I also give other advice. If I find a question, I check the current state of the dialog. Sometimes, I can directly suggest possible answers based on the information that is there. Mostly, I ask a few clarifying questions and then get back later. I also like to check in on dialogs I have participated in earlier and ask further follow-up questions or give answers, but I don't have to—if I don't continue, others will usually take up the slack.

It's enjoyable because it's good practice for me, I know that I'm helping others, and I get paid if I make helpful contributions—even if I can't directly answer someone's question, which is the case most of the time.

I don't repeat myself as often as you'd think; I think there is some sort of automation that reuses my previous answers in future dialogs.

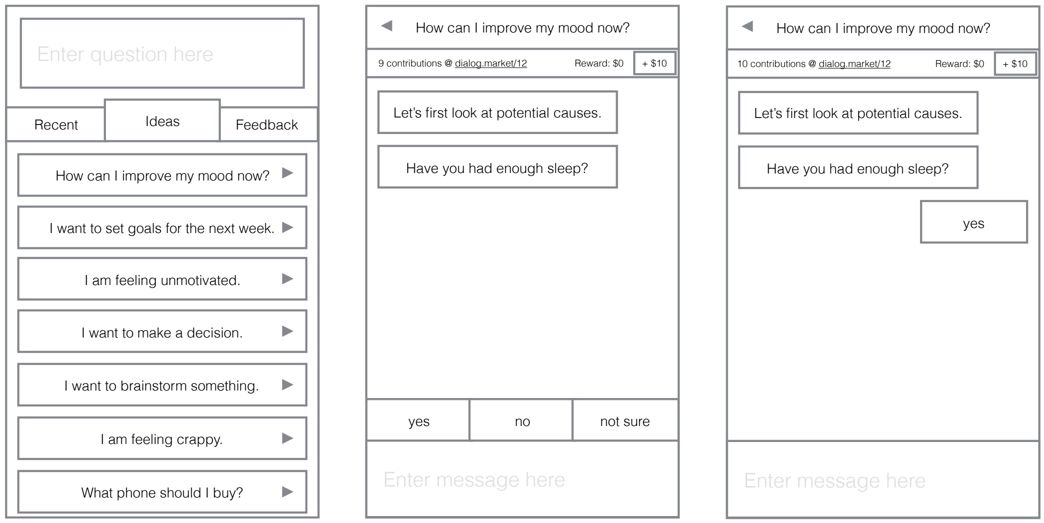

Figure 1: Askers use a mobile app to start dialogs, answer follow-up questions, and receive responses.

Actually, most of what I do now is judging the quality of others' questions and responses. This is even easier than writing responses, and I can do it from my phone. It gives me a question, some context, and a followup question or answer, and I give a rating 1-10 whether I think that it's a good answer, and what I think other people will think about it."

3.3 A programmer working on automation

"You know that site where you can get paid for contributing to dialogs? You don't have to do it in person—you can also write programs to do this. If your programs produce good contributions to dialogs, you collect the same reward you would as a human contributor. They make the data for basically all dialogs available, in anonymized form. I've been writing programs that use data on past dialogs to automate parts of new dialogs.

Initially, I wrote programs that simply tried to find very similar existing dialogs and then asked follow-up questions that were asked and judged useful in earlier dialogs. This worked well for a while, but there is a lot of competition in that space now. Other people are writing programs that are way better at deciding whether two question strings have the same meaning than the simple bots I wrote.

Now I'm focusing on writing bots for specific domains. I've had good success with my med bot. I looked at a lot of medical dialogs, checked with text books, and my bot now successfully narrows down diagnoses for some symptoms. Unfortunately, a lot of people are working in this space nowadays, too, and they have better access to actual doctors than I do; I heard that there will soon be an entire company focusing on medical dialogs."

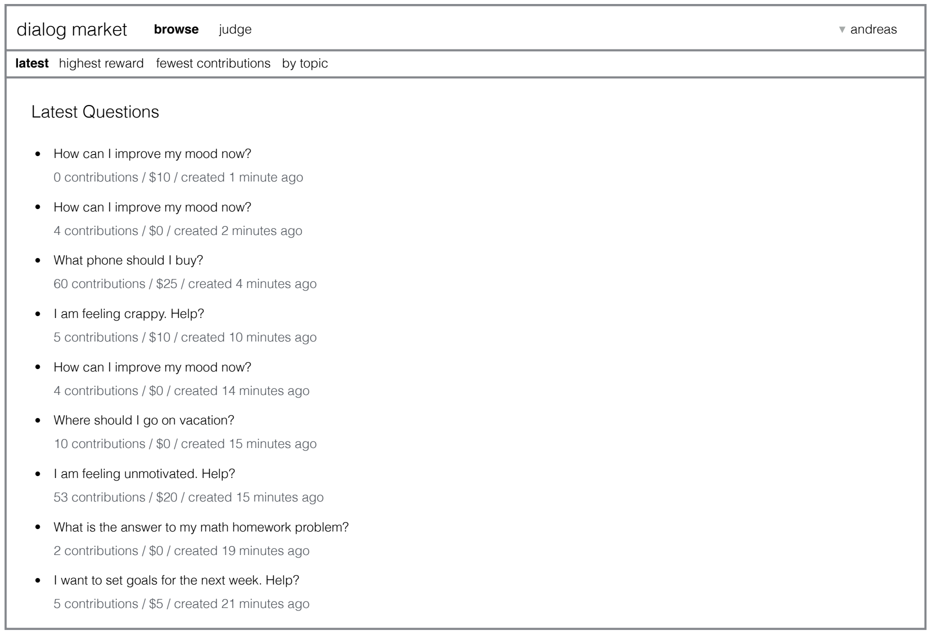

Figure 2: Contributors browse dialogs on the web.

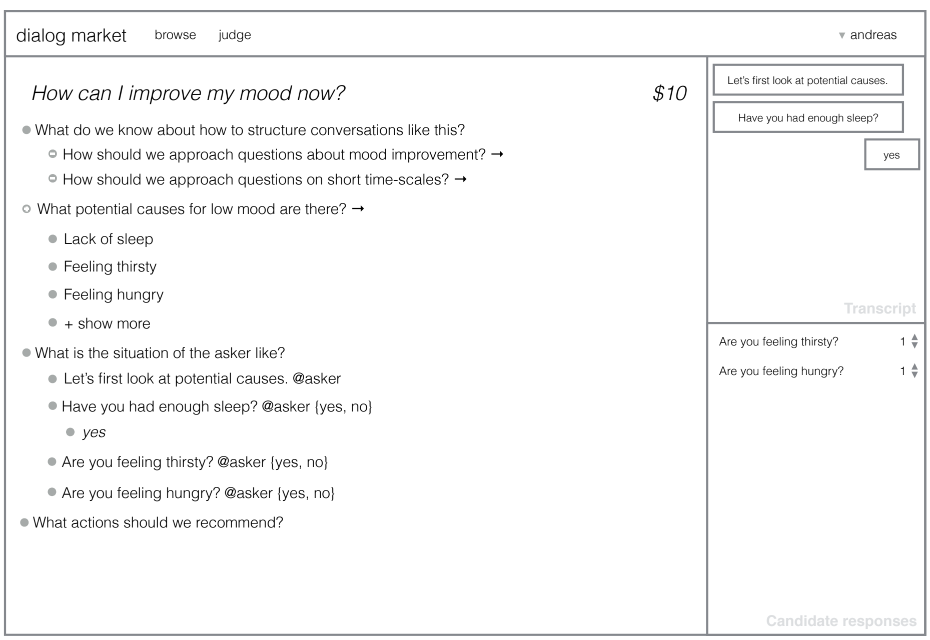

Figure 3: Contributors view a dialog on the web. On the right, we see the sequential conversation with the asker (top) and a few candidate next responses (bottom).

4 The System

I will now describe the architecture of the dialog markets system. I will first talk about sequential and tree-structured dialogs, the different roles they play, and how they can be integrated. Shortly after, I will discuss how to incentivize high-quality contributions to dialogs. To facilitate this, I first need to talk about what I mean by a "contribution". I will then introduce the reward distribution problem, but mostly postpone its discussion to Section 5. Finally, I will describe a set of components, including user interfaces, that together could form a practical dialog market.

4.1 Dialogs

A dialog is initiated by a person (or a program) who I will call the author (or asker). A dialog starts when the author asks a question. The goal of the dialog is to solve the problem that prompted the author to start the dialog.[1]

4.1.1 Sequential conversations

When I hear "dialog", my first association is with one-on-one conversations that proceed in a sequential manner, with each of the conversation partners taking turns. In the context of this project, I'm particularly thinking about text-based dialogs where each contribution is a short chat message, and where each participant can "append" a new response to the conversation at any time. This is a natural mode of interaction for most people. If someone asks a question, then this mode is well-suited for asking follow-up questions, receiving clarification, and providing small amounts of information. At each point in time, there is a clear context for new responses. If a question is asked, it is generally up to the other participant to either answer it or reject it by changing the topic.

4.1.2 Tree-structured conversations

While sequential dialogs are well-suited for transmitting information between two people, they are less well-suited for communication within a group, and also less well-suited as a means of knowledge representation. Within a group, we would like to keep track of multiple discussions at once. In this setting, hierarchical structure helps: it lets us navigate to the desired sub-discussion more quickly and allows different people to discuss different topics simultaneously. Tree-structured comment threads in particular have proven useful for discussions on the web. In this project, we're particularly interested in tree-structured discussions where each comment is either a follow-up question or a response to a question. We can generalize the traditional setup slightly by allowing graph-structured discussions, i.e. the same sub-conversation can be included in multiple places within and across dialogs.

4.1.3 Integrating sequential and tree-structured conversations

In our case, we need to structure the interaction between the asker and a group of answerers. We'll consider sequential dialogs for the interaction between the group of answerers (treated

[1] The initial question doesn't play a big role. Suppose the author submits an empty question. The market can still ask follow-up questions like "What would you like help with?", and propose specific ideas if the asker can't come up with any. As long as the asker can tell (with help by the market) that some responses are better than others, a helpful dialog can result.

as a single entity) and the asker, and tree-structured dialogs for the interactions within the group of askers and as a means of knowledge representation.

How do we integrate sequential and tree-structured dialogs? As a first step towards integration, imagine that we allow contributions in the tree-structured dialog to be sent to the asker's sequential dialog. This could happen by marking such contributions with `@asker` or `@author`. However, this may lead to chaos, since multiple questions may be posed nearly simultaneously, and since questions won't always fit into the flow of the sequential conversation. One way to address this is to *not* send contributions `@asker` directly to the sequential conversation, but rather to a list of candidate responses. This allows the system to decide more systematically which of these candidates to send next, based on various signals (e.g. upvotes that are reset whenever a new response is posted to the sequential conversation).

Since the tree-structured dialog also serves as the repository of knowledge about the current state of problem-solving for the asker's question, it's not enough to allow users to add to it. We also want users to be able to make edits, deletions, and structure changes, analogous to how you might edit a Wikipedia page. This changes the dynamics somewhat: when users add to the tree-structured dialog, they don't "own" what they add; others can change it when they see room for improvement.

4.2 Contributions

4.2.1 Questions and answers

Let's talk briefly about what individual elements in the tree-structured dialog may look like. There is a single top-level question. All other elements are either follow-up questions (to the top-level question, or to another follow-up question), or responses to questions. A response is not necessarily a complete answer. For example, if the question is "What are some potential causes of the flu?", a single response could be "Influenza virus type A".

Questions can restrict their answer format. This is most useful for questions directed at the asker, since we would like to minimize how much work they need to do. For example, there can be binary, multiple-choice, fill-in-the-blank, and free-form questions. We can also imagine more complex answer formats, such as files that need to be uploaded, camera pictures that need to be taken, or locations on a map that need to be selected.

4.2.2 Arbitrary dialog actions

In the next section, I'll talk about how we want to reward contributions based on how helpful they are. This will require that we know what we mean by a "contribution". As mentioned above, in addition to adding content to a dialog, there are other actions that users can take to modify the state of the dialog: They can make edits and structure changes, delete content, propose to send a notification to the asker, perhaps upvote such a proposal, perhaps propose to pay a reward to some user, and there are probably other actions that I am not thinking of right now. So, when I am talking about a "contribution" in the following, I will refer to the set that includes all such actions that change the state of the dialog in one way or another. This is comparable to making changes to a set of files and directories under git version control: each commit is an action.

4.3 Incentives

4.3.1 Associating rewards with dialogs

When the asker starts a dialog by posing a question, they can also pledge a reward. (I will primarily be thinking about monetary rewards, but we could also replace this with another quantifiable, transferable source of value, such as magic Internet points.) The role of the dialog market system is to distribute this reward to (the authors of) contributions based on how helpful they are, in order to incentivize dialogs that are most helpful to the asker (discussed in the next section). Pledges draw the attention of answerers to dialogs with the highest rwards. The asker (and other users) may also be able to pledge additional rewards for sub-questions in the tree-structured dialog in order to encourage additional work on these questions.

4.3.2 What are rewards used for?

How can we distribute rewards in a way that incentivizes dialogs to be as helpful as possible? There are at least two ways in which the system can use rewards:

- Rewards can be distributed directly to users who supply good contributions of any kind (useful follow-up questions, helpful edits, informative votes, etc.).
- Rewards can pay for the work necessary to determine which contributions are good. This can happen through auxiliary questions that are asked within the system ("How good is the following question/answer within its context?"). Note the recursion—we can ask the meta-question about answers to this question as well, but the available resources diminish quickly.

4.3.3 Desiderata for reward distribution

I'll discuss reward distribution strategies in more depth in Section 5. For now I want to cover some general properties that we would want this use of resources to have, ideally:

- **Byzantine fault-tolerance:** Even if some participants (people or programs) act maliciously—e.g. by providing unhelpful questions, or intentionally incorrect answers—we would like the overall system to perform its function undisturbed. This includes a number of special cases:
  - **Spam resistance:** As a special case of fault-tolerance, we would like the system to be resistant to spamming of unhelpful responses. One way to accomplish this is by making all contributions costly, under the assumption that participants will pay because they expect their contributions to pay dividends over time. If every participant pays a small fee for their contributions, then low-quality contributions (if present) could subsidize high-quality ones.
  - **Manipulation resistance:** Suppose I ask the question "Should I go to conference z?", and suppose that, under reflection, the arguments for and against are roughly equal in strength. The organizers of the conference have incentive to supply arguments in favor. If such arguments are provided in response to a subquestion "What are arguments in favor?", then no conflict of interest is present. On the other hand, if the organizers then provide an affirmative answer to the overall question, pointing to the arguments in favor, the judgment

is biased. We would like to set up incentives such that the overall mechanism is incentive-compatible, i.e., such that every participant is best off reporting their true beliefs.

  – **Stability in the presence of high rewards:** We would like the system to scale to scenarios where some questions have very high rewards, so it is important that the mechanism cannot be gamed.

- **Allocate money and time in a way that reflects what people care about:** If a subquestion is part of many dialogs, we want a lot of money to flow to this question in order to direct participants' attention towards its solution.

- **Sane temporal dynamics:** Contributions that turn out to be good later, but don't look good now, should still be rewarded eventually. This could be accomplished by distributing money over a longer period of time. It is impossible to have most of the payoff in the future at all points in time, but we could use some (temporally) heavy-tailed way of distributing money, although that raises concerns related to human psychology and discounting. The economically sound way to do this would be something like a prediction market on future judgments, but this would likely introduce a lot of overhead.

4.4 Architecture

Let's talk about a system that could make use of the principles described so far.

4.4.1 A minimal core

A key design goal behind this system is that its core is minimal. To the extent that complex mechanisms are necessary to answer questions, we would like to offload this work as much as possible to market participants, such that these mechanisms can be subject to improvement under competitive pressures over time.

For example, one can imagine sophisticated knowledge representation mechanisms for automating parts of dialogs. We would not want such structures to be part of the core system.

Similarly, there are many specific techniques we could use to improve the quality of contributions in this kind of system (e.g., Eigentrust, Bayesian optimization, Bayesian truth serum, ask about counterfactuals, adversarial setups, costly identities + reputation). My hope is that we can find a core market mechanism that establishes the right incentives, so that adding other techniques (via bots, see below) is profitable if and only if it improves the quality of dialogs.

I will now talk about the different (central and less central) components of the system: there is the dialog market itself, the reward computation system, user interfaces, and automation through bots (Figure 4).

4.4.2 Dialog market server

The dialog market server manages user accounts, the creation of dialogs and contributions, and incoming and outgoing payments. The distribution of payments depends on the reward computation system. The dialog market server is simply an API endpoint, i.e., it only talks to other machines; there is no user interface. This is the primary point of interaction for people who write bots.

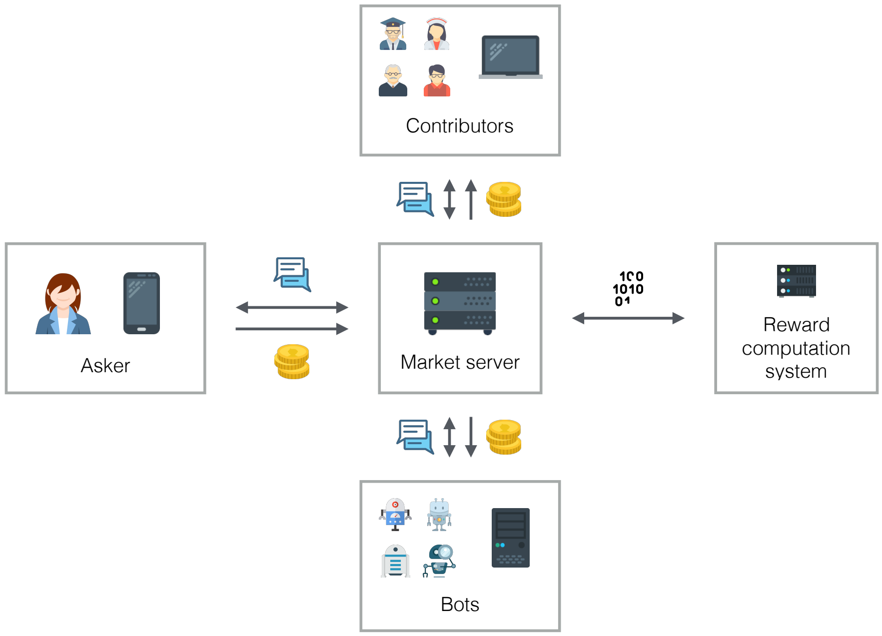

Figure 4: The dialog market server orchestrates the interaction between the asker, contributors, bots, and the reward computation system.

4.4.3 Reward computation system

The reward computation system takes as input a pointer to a dialog and (control of) a budget for judging the contributions in this dialog. It returns (eventually) an answer to the question of how to divide up the total reward over individual contributions. This system is also just an API endpoint and doesn't interact with users directly. It depends on the dialog market, as it may need to create dialogs and sub-questions to elicit information that is needed to compute rewards.

4.4.4 User interfaces

I can think of two user interfaces that would be useful.

First, a *website* could provide a user-accessible view of the dialog market. It could provide a way to browse and search dialogs, and to contribute to dialogs. This could be the primary point of contact for people who answer questions, make edits, ask follow-up questions, etc. For example, the website could look as shown in Figures 2 and 3.

Second, a *smartphone app* could be the primary point of interaction for users who seek to have their questions answered. The app could be used to quickly submit questions, to browse answers and follow-up questions, and to answer follow-up questions directed at the dialog author. The follow-up questions would be encouraged to be easy to answer on a smartphone, e.g. by making them multiple-choice. The app could then show regular updates on a user's questions, requiring only small, easy inputs now and then to enable answerers to make progress. Users could get push notifications when new answers or follow-up questions are available. The app could look similar to the one shown in Figure 1.

4.4.5 Automation using bots

Finally, there are programs that interact with the dialog market server. I'll call these programs bots. This component is not strictly necessary, and may to a large extent be developed by people outside of the organization running the market, but I expect that it will play a key role.

Initially, I expect almost all answers to come from humans. Over time, some answers and follow-up questions will repeat. A simple bot could replay contributions that were previously useful in the same context. The notion of context identity used here could be expanded over time using natural language processing tools. In this way, the system supports incremental automation.

Over time, this could result in good decision trees: bots could automate more and more contributions until the entire process of solving common problems is automated except for border cases. In other words, after some time, no human intervention would be necessary to solve the most common problems; only when we get to the fringes of the decision tree would human intervention be necessary.

Other possible bots include a Mechanical Turk bot (that simply outsources questions to Amazon's Mechanical Turk), and simple pattern matching bots (e.g., for Boolean questions such as "Is it the case that x?", it could ask the subquestions "What are arguments that x?" and "What are arguments that not x?").

5 Reward Distribution

5.1 The reward distribution task

Recall the setup: a person asks a (possibly very vague) question, which starts a dialog. This person offers some monetary reward that goes towards funding the dialog. Imagine the dialog as a tree of short contributions. Contributors—both humans and automated systems—add follow-up questions, comments, and partial answers to this tree. The goal is to produce a dialog that is as helpful as possible for solving the problem that prompted the person to start the dialog. The task that the dialog market faces is to distribute the reward over these contributions in proportion to their helpfulness, such that helpful contributions are incentivized.

This reward distribution task cannot be left (only) to the original asker. First, they will not always be in the best position to judge how helpful contributions are (e.g., consider medical advice, where they don't have the required knowledge; or dialogs with multiple partial solutions where the problem of assigning causal responsibility for the overall solution is itself a difficult problem). Second, I anticipate that some dialogs will be very large (thousands of contributions, say), with subdialogs that don't involve the original asker much, and it would be burdensome to require the original asker to judge all of them.

At the same time, the distribution of reward needs to be grounded in the original asker's values; a contribution is only helpful to the extent that it contributes to producing a dialog that is helpful for this person.

So, how do you distribute reward over contributions in a way that reflects the original asker's values? This is the reward distribution problem.

5.2 Redirecting meta-level questions to the dialog market

The reward distribution problem can be viewed as just another set of questions ("How much reward should be assigned to this contribution?"). Hence, it is natural to wonder: can we outsource questions like this to the dialog market as well?

This is appealing, since it allows arbitrary considerations to be taken into account when judging contributions and assigning reward, and since any improvements to the object-level system also apply to the evaluation mechanism.

Generating judgments through argumentation can be much better than pure voting. Votes provide very little information (a number + author identity + what is being voted on, and when)—it can be difficult to tell, for example, whether two votes are given for independent or redundant reasons, which makes aggregation challenging. Arguments, on the other hand, can more easily be judged on their own terms, and combined in a way that is sensitive to their content.

Therefore, it seems desirable to solve the reward distribution problem by redirecting meta-questions such as "How helpful is this contribution?" back to the dialog market, and to answer them using the same mechanisms that are used to answer other questions.

Of course, this cannot happen for every contribution (as this would result in infinite regress). So, we might only investigate with some probability, or ask more general questions such as "How helpful are Bob's contributions in this dialog?".

5.3 The grounding problem

Redirecting meta-level questions to the market raises the grounding problem: if the dialog market works, then meta-level arguments and judgments will be good, which leads to good market incentives, and therefore to the overall system working; if the dialog market doesn't work, the meta-level judgments will be bad as well, and the market will incentivize the wrong kinds of contributions.

In other words, there is a chicken-and-egg problem: to get a working dialog market (with respect to object-level contributions), you already need a working dialog market (with respect to meta-level contributions).

5.4 Strategies

I don't have a satisfying solution the grounding problem yet. I will first list some general directions that seem promising, and then two concrete approaches, but there is still a lot of room for improvement.

5.4.1 Start with what can be relied on

Any solution to the grounding problem will have to make use of some information sources, and will assume that they satisfy certain criteria. One way of approaching the grounding problem is to ask: what information sources are there? We need to be careful about incentives—sources that are usually reliable may become unreliable in the presence of market incentives.

Here is a list of some information sources:

- I (as the system designer) will use my best judgment about what is helpful. (Of course, my best judgment in any given moment may not be very good.)
- I trust certain market participants to use their best judgment (to some degree). This trust can be topic-specific, and transitive (to some degree).
- The original asker will use their best judgment when they rate contributions or answer follow-up questions. This is no longer true when there are dependencies between dialogs, e.g. through reputation effects (Bob might answer his own questions using sockpuppet accounts, and rate these answers highly to increase the probability of getting high reward for future answers).
- When people bet substantial amounts of money, they will bet according to their best judgment, unless the side-effects of betting otherwise outweigh the gains from winning.
  - Some answers can be verified, which makes it easier to judge them, and therefore to judge others' judgments:
    - Some answers can be checked empirically.
    - Some answers can be checked logically.
- We can require answers to be given in a form that is easy to judge.
- When identities are expensive, participants will avoid actions that lead to identities becoming useless ("destroying their reputation"), unless benefits outweigh costs.

- We can probably make some assumptions about contributions in general. It is unclear what exactly they are ("there is some signal in some contributions, probably"), but an adaptive system may be able to learn this over time.

5.4.2 Combine many weak heuristics

In judging contributions, there may be many considerations that can be relied on weakly (e.g., reputation, agreement between raters), and no single information source that can be strongly relied on. However, this still raises the question of what the process looks like that integrates these considerations. Ideally, these considerations can come in the form of arguments in the dialog system. However, this brings us back to the start: how do we integrate these arguments, and judge which are good? For any particular way of integrating them, how do we have assurance that it will lead to good judgments? Can we build a reward distribution system that learns over time how to combine such heuristics?

5.4.3 Defer to more trusted systems

When we have different evaluation methods, or multiple information sources, and it is costly to evaluate in depth, we can choose to only evaluate in depth sometimes. This choice can be made randomly (defer with some probability) or based on discussion features (e.g., defer when there is disagreement). The results of such deferral can be taken into account by reputation systems (if, whenever we evaluated in depth, Bob's judgments turned out to be good, we might want to rely on them more in the future).

A tempting option for deferral is to increase the reward with some probability; e.g., with probability p, offer $\frac{1}{p}$ reward, with some upper bound. However, this already assumes that the dialog market works when rewards are high, so this option may not help with the fundamental grounding problem, but could be useful in boosting a partial solution.

Deferral to more trusted groups of market participants doesn't suffer from this problem, but also seems problematic on its own, as it could limit the overall capability of the system to what this trusted group can verify on their own.

A potentially promising idea is to combine deferral to more trusted participants and deferral to the market with higher rewards: defer to more trusted participants, but provide them with some (potentially large) amount of reward to spend on the dialog market. This allows the trusted participants to make use of any capabilities that may be present in the dialog market, but they may also ignore the market to the extent that it isn't useful. For instance, a trusted participant may ask the market to judge a contribution and to provide easily understandable justifications; then she can take these considerations into account to the extent that she understands them. As presented, this proposal seems like it would still significantly restrict the capability of the market (limiting ideal meta-level judgments to arguments that a single trusted participant can understand, augmented by the use of the market to help them in their understanding). However, there may be proposals of similar shape that are less restrictive.

5.4.4 Defer evaluation to the future

A particular type of deferral to a more trusted system is deferral to the future. As time goes on, we tend to collect more information that may help us determine to what extent contributions are helpful. For example, we learn more about whether advice helped and whether predictions came true.

One way to make use of this effect is to make payoffs temporally heavy-tailed: e.g, spend 50% of the reward for a dialog by the deadline given by the user, then 50% of the remainder after 2x this time, 50% of the remainder again after 4x, after 8x, etc. This could lead to dialogs becoming better over time, which allows better judgment of how good the early contributions were, and which thus incentivizes good early contributions.

Reputation can be used to amplify the effects of the small residual payments. If Alice's judgments tend to turn out to be good after many months or years of deliberation, we should probably trust and reward her more.

5.4.5 Grow slowly, check behavior at each stage

Finally, one can give a non-answer: simply implement a dialog market, restrict it to a particularly cooperative set of participants, see how well it works, grow it slowly, check for failures and degradation of performance at each stage, and apply sophisticated reward distribution strategies as they become necessary. This is a reasonable strategy, but I'd prefer to follow it in conjunction with more theoretical guarantees, not as the sole strategy.

5.5 Reward distribution using escalating bets

I will now give an example of how one might solve the grounding problem. This solution relies on people betting in their self-interest and on the original asker using their best judgment (informed by information provided by the dialog market).

There is a lot of room for solutions that combine the ingredients given in the previous section in different ways. The particular approach I'm showing here mainly serves to illustrate that this is indeed a solvable problem. It is directly based on ideas by Paul Christiano (Of Arguments and Wagers; personal communication).

Before I go into details, here is an overview of what happens from start to end in a dialog under this proposal:

1. **Start:** The asker—we'll call her Alice—starts a dialog by submitting a question, reward, and expiration date/time.
2. **Contributions:** Alice and other users (bots and humans) post follow-up questions and other responses to the dialog, make edits, and contribute in arbitrary other ways.
3. **Bets:** For each contribution, users can bet on how much Alice would want to pay to the contributor, under reflection. The stakes for these bets are fixed and initially small.
4. **Escalation:** At expiration time, the system checks all bets, and escalates ones with disagreement, potentially all the way up to Alice.
5. **Conclusion:** When all bets are settled, either by agreement or by Alice's decree, a single best answer has been selected for each of the reward questions. The system enacts these conclusions by paying out the corresponding rewards.

I will now talk about some of these stages in more detail.

5.5.1 Betting on reward choices

For each contribution, users can bet on how much Alice would want to pay if she thought about it (or, more precisely, what payment will be selected by an escalation process that involves increasingly well-funded dialog markets investigating what to select, and that potentially ends with Alice making a decision; see below).

Bets happen simultaneously with other contributions. A little while before the dialog expires, we stop all contributions. From this point on, users can only update their bets.

Users can edit their bets at any time before the expiration time.

Adding a contribution may automatically set up a small default bet by its author on the corresponding payment question, so that (1) other users are incentivized to bet, and (2) to disincentivize spam.

5.5.2 Escalation on disagreement

For contributions where the bets don't indicate substantial disagreement about how much to pay, we determine the payment using the data provided by bets.

For contributions where bets do indicate disagreement, we escalate. This means:

1. We increase the stake required for the bets, and set a new expiration time for the bets
2. We implement arbitrage on the bets, and use the guaranteed profit to:
   - set up a system bet for the next round (to incentivize participation)
   - fund a dialog market that investigates the question: "How much should Alice pay for [contribution]?" (if the profit exceeds some threshold)

We repeat steps 1 and 2, incrementally increasing the stake. If the bets indicate agreement at any step in the ladder, the process stops and all bets are settled.

For each stake size, we compute how many users are eligible to participate in the bets. As the stake grows, this number shrinks. If this number is very small, we directly ask Alice to pick the answer she likes best, using the dialog markets funded by arbitrage profits as a source of advice. This settles all bets.

5.5.3 The end of a dialog

When all bets are settled and the dialog concludes, market participants have selected the payment that they expect Alice to choose if she thought about the choice and used the dialog market as a tool for investigation.

5.6 Reward distribution using probabilistic models

I will now describe a second approach to solving the reward distribution problem. This approach doesn't directly support meta-level discussion, and so doesn't run into the grounding problem, but is also less powerful than approaches that do support meta-level discussion. I am including it here as an example of a less complete solution that may be more practical early on.

This approach is fairly independent of the dialog setup, and so we if we build it, we could also apply it in other scenarios where we'd like to set up incentives for behavior that's good under some expensive/long-term metric, where we can measure many indirect proxies of goodness, but only get a few high-quality measurements, and where we want to incentivize people to make these proxies as informative as possible. For example, this sort of procedure could be used to distribute rewards to the authors of commits or pull requests in a git repository.

5.6.1 Elicit cheap, uninformative signals at runtime

While the dialog is happening, we elicit easy-to-provide information (such as upvotes/downvotes, likes, 1-5 star ratings) from the original asker and from other contributors. (This sort of information doesn't need to have special status within the system—it could be implemented via replies.)

This information is cheap to get, but it may reflect long-run helpfulness only in very indirect or noisy ways, or sometimes not at all. It will also generally not directly speak to the usefulness of structure changes, edits, and deletions. We can only make very weak assumptions about how this information relates to the long-run helpfulness of particular changes, at least initially.

This information also serves to guide the dialog while it is happening. For example, if the original asker likes a contribution, other participants may take this to indicate that they want to encourage similar contributions.

5.6.2 Elicit expensive, informative signals after the fact

A while after the dialog has concluded, perhaps after the asker has taken a decision recommended within the dialog and observed its outcome, we require the asker to answer questions to help us determine how much to pay for each contribution. We have some flexibility in what questions we ask here ("How helpful was the dialog overall?", "How much should we pay for change x?", "Was change x more helpful than change y?", "How much worse would the dialog have been if change x hadnt́ happened?", etc.), but we will be limited to a relatively small number of questions. We definitely cannot ask about each contribution.

In contrast to the questions we asked while the dialog was happening, we will make strong assumptions about how the questions asked here relate to long-run goodness, and thus to how much we should pay. In other words, we assume a mostly fixed semantics for these questions. As an extreme case, we could simply ask "How much do you want to pay for change x, knowing what you know now?" for a few contributions and take that at face value. (However, this might be a particularly difficult question for people to answer, so other questions with comparably clear semantics might be preferable.)

5.6.3 Learn a model that relates cheap and expensive signals

Since we can only ask a few questions in (2), we need to determine (a) what questions to ask and (b) how much to pay for contributions that the asker doesn't directly evaluate. To this end, we will learn a probabilistic model (across dialogs) that relates the early/cheap pieces of information to the answers to post-dialog questions, and thus to facts about how much to pay per contribution. For example, we might learn that when Bob indicates that he likes a contribution in dialogs related to where to go on vacation, it will frequently turn out that the asker rates it highly in retrospect.

Given this model, we ask the post-dialog questions that, in expectation, most reduce the variance of our distribution on how much to pay (within this dialog, or taking into account future dialogs). We pay contributors based on the expected payments under this model.

5.6.4 Incentivize cheap signals to be informative

We don't just want to reward people for helpful contributions, but also for helping us figure out how much to pay by providing the information in (1). I don't know what approach to choose here, but it could be something like taking a constant fraction of the dialog reward, and paying each info provider in proportion to the information gain between the distribution on payments excluding their data point, and including it (e.g., with and without taking into account the fact that Bob liked a particular contribution); or before/after their data point.

5.7 Summary

It is appealing to consider dialog markets that solve the reward distribution problem by redirecting meta-level judgment questions back to the market. This raises the grounding problem: the quality of judgments will only be good if the market works, which in turn depends on good judgments. To solve this problem, it seems necessary to apply some additional techniques, such as relying on people's self-interest when betting or relying on the original asker's judgment. I have described a candidate for a reward distribution system that attempts to solve the grounding problem, but haven't given any serious arguments that it will work, and so expect that (in its current form) it will run into issues. I have also described another approach to reward distribution that is less powerful but perhaps more practical initially. Overall, the question of what reward distribution systems work well in practice is still wide open.

6 Applications

In the following, I will describe a few possible applications. I don't claim that dialog markets are appropriate for all of these; I mainly want to illustrate that there are many candidates for applications, so our approach are probably appropriate for some, and there is some indication that this may be a useful domain-general tool.

6.1 Making domain expertise available

Consider domain experts such as lawyers, doctors, and tax accountants. Dialog markets could provide a lower-friction way for experts to turn their knowledge into money (although domain-specific restrictions apply, as is particularly clear in the case of lawyers). Dialog markets could also automate the easy, repetitive parts of experts' jobs. If certain questions always result in the same follow-up questions, we can detect this pattern and then ask the appropriate follow-up questions automatically.

6.2 Knowledge transmission within organizations

Dialogs could answer questions that employees in big companies have about the workings of the company. Similarly, the management of that company could elicit information from the company at large (e.g., "Do our customers in Asia like our new product?"). Instead of working down the command hierarchy, they could directly pose the question to the company and assign bonuses for helpful contributions.

6.3 Self-help

In a different project—the HelperBot—I explored whether an automated question-asking system could be useful for thinking about various issues in one's life. My impression is that it can be somewhat useful for some people, but that good prioritization of questions to ask next requires human ingenuity, or at least human feedback, and that, more generally, we may not know enough about knowledge representation yet to do this with much success. By contrast, in my (and others') experience, dialogs with other people often do seem to help with tough problems. In part, this is because they help us get over activation thresholds—it is easier to answer questions than to push the investigation forward on one's own.

As a particular instantiation of this application, we could consider "semi-automated cognitive behavioral therapy" where machines ask some follow-up questions.

6.4 Product recommendations and ads

People often look for good products and services. For example, I have recently been thinking about whether to buy a car, and if so, which one. The right answer depends on what exactly my requirements are, so it is important to ask me questions to figure out what car might be appropriate for me.

Suppose that putting up answers has some monetary cost. If the incentive system is set up appropriately, then people could put up advertisements without cost only when they expect them to be helpful to users. People could put them up when they are not helpful, in which case they would subsidize the other participants of the system by paying a fee that they don't get back. Of course, figuring out an incentive system such that dynamics like this emerge could be quite tricky.

6.5 Various kinds of personal advice

There are existing services that provide personalized advice. For example, there is personalized voting advice (automated e.g. in the "Wahl-o-mat" in Germany) and personalized career advice. The system could take in the fixed decision trees used by existing tools (through bots) and then incrementally improve their answers through properly incentivized dialogs.

As another example, we can imagine personalized advice on what charity to give to.

6.6 Brainstorming

We can also think about situations where we (as the asker) are looking to generate a large number of solution candidates, and then will pick a single answer ourselves. This could happen within the dialog system by restricting the answer of the top-level question to the author of the question, allowing the general public only to add subquestions and -answers.

For example, there are existing websites that offer community-driven brainstorming of company names in exchange for a fee; this could be done as a special case of the more general dialog system described here.

6.7 Argument mapping and checking

Suppose I am interested in the current state of the debate on whether humans are causing global warming. I could put up this question, pay a price towards it, and then have others flesh out the debate for me.

The system may even be useful in cases where one doesn't expect others to do much work in answering the question per se, but simply wants others to check one's arguments. As a researcher, I might want to ask one of my research questions, and then add subquestions and answers that reflect my current understanding of the topic, in the hope that others will ask helpful questions, or will help me make progress by giving feedback on my current thoughts. Think about it as a structured editor for thoughts, with help by others in proportion to how much money you pledge. If you think your own thoughts are very high-quality and complete, you could pledge a large amount of money on your dialog (well, monolog) once you have fleshed it out. Then others would be incentivized to judge your argument with high precision, and to find places where your argument "leaks money", as that would be an opportunity for them to earn part of your pledge.

Given a large-enough public repository of arguments that make money from questions that they are answering, you could imagine people (or, more speculatively, programs) scavenging for flawed arguments, as those are an opportunity for redirecting the flow of money towards the program's owner if they can fix the argument.

6.8 Referral to existing tools

Some questions can be answered without dialogs. For many questions, a referral to existing tools (e.g., web search, route planning, Quora, StackOverflow) might be all that is required. It is easy to imagine bots that detect some of these cases and that provide links to answers given by existing tools. Indeed, if the system described here were to achieve some popularity, the owners of existing tools would have incentive to provide their answers within the system to make money and/or get referrals.

6.9 Explanations & teaching

When we first learn about a topic, our background knowledge determines what constitutes useful information. It would make sense for a learning system to ask a few questions to determine what we already know before it presents information. It could also ask questions to check our understanding.

This is feasible with human tutors, but would scale much better with bots that have more sophisticated knowledge representation than what is possible using current AI techniques, so nontrivial versions may be off some ways into the future.

6.10 Research funding

On a larger scale than previous applications, funding agencies could pledge rewards on general questions such as "How does brain work?". Researchers could then ask and work on relevant subquestions. This has some advantages over the current journal- and grant-based system: small, incremental contributions are made easier and are rewarded more proportionally, especially for non-academics; negative results may be easier to publish if they are clearly relevant; and it may be easier for researchers to learn about the current state of investigation for a particular question.

7 Discussion

7.1 Why this might be a good idea

This is relatively feasible. (1) This can be kickstarted using Mechanical Turk. A bot could post questions to Mechanical Turk and retrieve answers. This provides a (possibly very bad) lower bound on answer quality, and upper bound on response times. (2) Unlike many other conceivable proposals related to knowledge representation and reasoning, it doesn't require magic, i.e., it doesn't require machine reasoning and knowledge representation beyond what is possible given the current state of the art. Instead, we can incrementally move towards automation as better natural language processing and knowledge representation tools become available.

This shares some features with successful companies. (1) It is plausible to me that cognitive resources are currently underused. A number of companies, including Uber and AirBnB, have built their success on making illiquid resources more liquid. Cognitive resources could be such an illiquid resources. (2) There is a clear business model: keep a fraction of each transaction as a marketplace fee. (3) There is an additional business model that stems from being a market participant (automation using bots). Together with (2), this follows a pattern of building infrastructure and then being its primary user, a pattern that Amazon has followed. (4) There is a huge market, ultimately, as illustrated by the myriad of existing tools that are special cases.

This could be substantially better than existing tools. It could be more compositional, with better incentives, more meta (redirecting part of the reward distribution process to the system, and optionally having discussions associated with it), more personalized/context-sensitive, more automated, and with lower activation threshold for asking questions (the initial question can be short and vague, in contrast to the well-formed, complete questions required by existing systems).

In the long run, this has the potential to create good jobs. People could earn money under good conditions—working from home, at arbitrary times, doing tasks that are fun and intellectually stimulating, and personally rewarding because people know that they are helping someone. People could specialize in particular topics (or are already specialized), and could make money using their specialist knowledge. Others could write bots that use existing answers to suggest answers to new questions. In the beginning, basically all work would be done by humans, but over time, opportunities for automation would be discovered. People could create a passive income stream if they created good answers (or, even more so, bots) that persist over time and that answer questions that many people ask. Companies could be built on top of this system; e.g. a company could offer a simple decision tree builder that uses lightweight natural language processing and that allows people to build domain-specific bots without programming experience.

This project is *sufficiently meta.* If the basic system works (i.e. if it is useful, even if far from perfect), we could outsource parts of the task of improving the system to the system. For example, we could have a dialog that discusses the rating system. More generally, dogfooding—using the system in the process of building the system—could be useful for two reasons: (1) A problem with many existing institutions is that they are insufficiently meta. Problems don't bubble up to the top, good ideas go unused, and there is too little reflection on what is going on and what should change. This is at least in part because no good mechanisms exist for aggregating such knowledge. Organization-wide use of a dialog system like this could potentially help avoid this fate. (2) Using the system every day would make it less likely that one builds something that no one wants.

7.2 Open questions—or, why this might not be a good idea

How long/complex can individual questions and answers be? We want questions and answers to be short to encourage compositionality. Instead of giving a lot of detail in a single answer, we would like such detail to be distributed into subquestions and their answers. For example, we could restrict answers to a single sentence, or to tweet size, or otherwise encourage short answers. What is a principled solution for this?

Would this be legal? Prediction markets are mostly not legal in the US. We would have to make sure that the precise mechanism chosen complies with the relevant laws. Some mechanisms that wouldn't work in public could still work within corporations (e.g., private prediction markets).

What about privacy? We want to share a lot of data between market participants for several reasons: (1) We want to share logs of dialogs to support the development of automated tools. (2) We want to re-use answers across dialogs. People may not want to ask some questions that they care about even in a pseudonymous setting, whereas they may do so if they are entirely anonymous.

What does a stripped-down version of this proposal look like? Building the entire system at once is probably too difficult. This raises the question which smaller system would be a good start. What is a minimal v0.1 that would still be useful? On a related note, what components does this proposal have that can be analyzed independently? For example, one could try to separately think about organizing knowledge in dialogs, about how to do an economically sound organization of a discussion forum, and how to accomplish spam/manipulation resistance using economic incentives.

Would response times be short enough? The non-instant nature of the system—i.e. the delays between asking questions and getting answers—could make it less fun to use. This isn't different from (e.g.) StackOverflow and Quora, but there are different expectations for chat-like conversations and forum posts.

Would money introduce too much friction? We don't want people thinking about money most of the time. This motivates having a default fee for most things as opposed to requiring a choice of payment amount at every step.

Would people be willing to pay? How expensive would dialogs be? People may be (perhaps unreasonably) unwilling to pay for soft things like knowledge that might feel difficult to justify. In effect, this is putting a lower bound on how reusable content needs to be (reducing the price per user), but this lower bound may be quite high.

What exactly should the incentive system look like? How do we validate it? Getting the incentive system right could be very difficult, and very important. At the same time, we wouldn't have to get it right immediately—it could evolve over time.

What exactly makes dialogs useful, and are we capturing these benefits? There are at least two seemingly separate components that contribute to the usefulness of dialogs. First, the dialog is likely to be more relevant for the original asker than other pieces of text, because it was produced in response to the asker's questions. Second, there are additional benefits due to the interactive nature of the generative process that produced the dialog: for example, if I hear a position on some issue, I might test it against three counterarguments that I randomly picked. If all are refuted by the person holding the position, I will have higher confidence that the position is solid than if the person holding the position had chosen the counterarguments to evaluate.

7.3 Comparison to existing tools

Google, in particular in combination with sites such as StackOverflow and Wikipedia, is great at answering factual questions like "What is the capital of Turkey?" or "How do I write a for-loop in JavaScript?". Dialog markets aim to do the same for questions that require personalized answers.

There are two main differences between dialog markets and existing tools that provide personalized answers (such as Quora, Yahoo Answers, Reddit, and other online forums):

First, the system is designed to **generate conversations that are as valuable as possible for the person asking a question**. On existing platforms, answers are written at least as much for other readers as they are for the asker. If I am writing for a large audience, I can't make my answers depend on very specific circumstances; on the other hand, if I am writing for a single user, I can ask many follow-up questions to make sure that I am really solving the underlying problem that prompted their question. To facilitate this focus on the asker's values, credit assignment is the core problem that dialog markets need to solve well. In existing systems, credit assignment frequently seems like an afterthought. By default, I also expect dialogs to be semi-private (non-searchable), so that the person asking can more comfortably provide personal information.

Second, dialog markets are designed to **allow for the incremental automation of contributions**. All design choices are aligned with this goal: We use monetary rewards, since reputation-based systems (as on Quora and Reddit) are unlikely to incentivize people to build substantial automation on top of the system. We aim for robust credit assignment based on the asker's values, since anything short of that can easily lead to low-quality contributions by profit-maximizing algorithmic participants. We favor individual contributions that are small, maybe single sentences, since it is much easier to automate such short contributions than to produce entire paragraphs that completely answer the question (as would be necessary in the case of Quora and Yahoo Answers).

8 Getting Started

8.1 Making things work on a small scale

8.1.1 The challenge

Dialog markets are intended to produce high-quality conversations that help users resolve challenging problems. One might worry that there is a mismatch between the difficulty of this task and the capability of the proposed architecture, in particular at the very beginning. The capability of the system depends on the relevant skill of participants and the power of the mechanism. These two factors can be traded off against each other to some extent, but if the overall combination is below some threshold, the quality of dialogs will be low, and in particular lower than what would be required for people to be happy to pay in exchange. So, it is important to understand how we can ensure that the quality of conversations is sufficiently high.

To illustrate the challenge, imagine that a crowd made up of Mechanical Turk workers was faced with the task of producing helpful follow-up questions and advice to the question "How can I improve my sleep?", without further assistance or filtering. The result would probably include some obvious follow-ups and ideas, but it wouldn't be the kind of advice that people would pay for. The advice wouldn't include much that people couldn't easily think of themselves (given that Turkers are in general not experts on the topic, are probably no better than others at using Google, and are less motivated than the person who initiated the conversation). While there are some potential advantages from having multiple people think about the problem and from getting outside perspective, there are also significant costs. Turkers won't know the asker's context, there is overhead in learning about it, and conversing with a group could be incoherent and could introduce delays. Overall, such dialogs wouldn't be worth much, most probably not enough to pay Turkers minimum wage.

In the next few sections, I will discuss some strategies that help overcome this challenge.

8.1.2 Incrementally grow the set of questions people can ask

The web-based system for answerers should be fully general, but for the (probably mobile) app for askers, it is probably better to incrementally expand coverage. Focusing first only on a single application, then on a few applications, and only later on expanding to arbitrary questions will simplify the problem and makes reuse/automation easier.

I am imagining a sequence of apps, for example: (1) "How can I feel better right now?", (2) "How can I sleep better?", (3) "How can I improve my relationship?", (4) "Should I go on vacation?", etc. I will discuss better sequences of initial applications in Section 8.2. For now, I am more interested in general properties of this sequence of apps.

The apps will all be essentially the same app, each with a different skin. There is a single button that starts a dialog with the corresponding question and then shows a sequential dialog view, allowing the user to provide multiple-choice and free-form feedback.

Depending on the app, we might choose a different revenue model:

- Optional tipping after the dialog is over (maybe using in-app purchases)
- Pay to continue the conversation after n exchanges
- First dialog free, fixed fee for future dialogs
- Affiliate links for movie/book/product/vacation recommendations

For each app, we measure how long it takes for the dialogs to be of high quality. The hope is that this time goes down as the system is supported by better automation and gains some general-purpose domain knowledge on how to approach certain kinds of dialogs. (I discuss accumulation of knowledge below.) The fact that we are ordering applications such that we tackle the easier, more concrete ones first will reverse this effect to some extent, but I hope that there will be a point—once the base of answerers is fairly broad and many topics have been explored—at which it takes much less time for the system to do well on new question types. When that time span is short enough, we publish the fully general app that allows people to ask any question.

8.1.3 Incrementally build up knowledge required to answer questions well

Share procedural and declarative knowledge across dialogs via "objective" subdialogs Answerers can create "objective" dialogs for questions such as:

- "How should we approach self-care questions?"
- "How should we approach career questions?"
- "How should we approach dialogs over short time-scales?"
- "What common causes of bad sleep are there?"
- "What are the most promising treatments for sleep apnea?"

These dialogs can be referenced/included within other "subjective" dialogs, which constitute a much larger fraction of all dialogs. A small part of the reward from these subjective dialogs then flows to the contributors of the included objective dialogs. If an objective dialog includes other objective dialogs, a portion of the reward flows to those as well, and so on.

I expect that there will be a division of labor, with domain experts mostly working on objective dialogs, which are very polished (maybe comparable to the best Wikipedia pages), and generalists doing some of the work on subjective dialogs, simply by following the prescriptions and processes outlined in the objective dialogs.

Adding references to relevant objective dialogs is a task that seems well-suited for automation.

Improve automated contributions over time via feedback mechanisms Since questions are initially chosen from a relatively small set, and since most responses are multiple-choice, it should be straightforward for us (as the organization running the market) to implement basic automation. When a dialog ends, we will elicit information from both askers and other participants on how well things went, and on how much (they think) different parts contributed to the overall outcome. This will help us preferentially re-use contributions that were most helpful in the past, or ones that are predicted to be most helpful in a new situation based on past judgments. (I'll discuss this approach in the section on reward distribution below.)

8.1.4 Curate the set of answerers

Seed with people who have skills appropriate to the task To generate high-quality *objective* dialogs, we may want to seek out experts in the domains we focus on. To generate high-quality subjective dialogs, I imagine that we will seek out people who have shown interest in, or skill at, careful reasoning and analytical thinking, but who are not necessarily domain experts.

I don't think the system can succeed if it is necessary to have domain experts run all dialogs. The required pay would be more than most people would be willing to pay. A key hypothesis underlying the dialog markets project is that a substantial part of the work can be done by groups of generalists as long as they have adequate reasoning skills and are equipped with expert-curated instructions. A further hypothesis is that a significant part of *that* work can be automated over time.

Maintain answerer quality using strict reward assignment To maintain high answer quality as the system grows, we use a strict reward mechanism, so that work that is below the target quality receives $0 (or potentially even negative) reward. This is intended to quickly disincentivize participation from workers who might otherwise swamp the system with low-quality contributions.

8.1.5 Filter what users see

By default, the sequential chat view (on mobile) shows only a small set of contributions to the original asker. The system will only send contributions to the asker if they are predicted to be helpful when evaluated later on (e.g. based on author, likes). At the very beginning, we may essentially manually curate what askers see. This allows the system to work well (from the asker's perspective) even if a significant number of low-quality contributions are posted to the web-based system.

8.2 Initial applications

Above, I outlined the strategy of selecting a *single* specific question as the initial application, then expanding to related questions one by one, until the system can learn to provide high-quality dialogs for new types of questions quickly. Here, I want to discuss what such a sequence could look like.

8.2.1 Criteria for good initial applications

We'd like to find an initial application that satisfies the following criteria:

1. *The problem is important to the target audience.*
   If people don't care about the problem, they will be less willing to try new approaches (e.g., they might not be willing to install an app), and will also be less likely to pay for helpful contributions.

2. *Our solution will be much better than the next-best solution.*
   If this isn't the case, there will be little incentive for people to switch over from whatever substitute they are currently using.

3. *The amount of information required from users is at the sweet spot between too little and too much.*
   If there is very little information required (e.g. for "What is the largest known prime number?"), we can't exercise our advantage compared to other Q&A sites. If there is too much context required, we can't provide high-quality advice.

4. *Simple automation can go a long way.*
   If each question requires a lot of custom human labor, we may still be able to produce a

market for conversations, but the prices may be higher than people would be willing to pay and we wouldn't be on a direct path towards building a market where automation can play a substantial part.

5. *We (as the system designers) understand the target audience well.*
   This will make it much easier to build a system that is likely to be successful, as we can get instant feedback by using our mental models of the target audience.

6. *There is a natural progression to other topics.*
   This is true for almost any question, but it is worth considering whether particular initial questions are likely to lead to better follow-up applications than others.

7. *We (as the system designers) are excited about the application and likely follow-up applications.*

The following criteria matter somewhat:

1. *We understand the domain well.*
   We don't need to be a domain experts ourselves, but we need to be able to judge how well the system is working, both from a user's perspective and from a domain expert's perspective. So, easy access to domain experts is a plus. In the ideal case, we can fill the role of the domain expert ourselves.

2. *People would be willing to pay for helpful advice.*
   This isn't strictly necessary for the first application, but should be the case for applications in the neighborhood of the first application. If it is unlikely that at least some people would be willing to pay, this may indicate that the problem isn't important to people or that our solution isn't good.

3. *People would want to use the system more than once.*
   This reduces user acquisition costs. However, this will also cause people to expect the system to know about and take into account previous conversations, so this may make it more difficult to reliably live up to users' expectations. For some single-use apps (such as "Should I buy or rent?"), we could allow users to go through a series of dialogs to decide ("Can I get a bank loan?", "Are there any reasonable apartments I could afford to buy?", etc.), making it effectively multi-use.

4. *The app doesn't paint a misleading picture of the longer-term goals for dialog markets.*
   In the beginning, it will take a lot of time to make an app work really well for any single question, so the first few questions will shape how potential collaborators view the project.

And here are some criteria that don't matter for the initial application:

1. *There is a large target audience.*
   The initial application only serves to seed the system, so doesn't need a large user base per se, as long as there is a trajectory towards applications with larger audiences.

2. *There is a natural way to make it viral/social.*
   This may matter for future applications, but not initially, where the main goal is to test whether our system can in fact produce high-quality interactions at low cost.

8.2.2 Potential applications

I brainstormed a list of about 300 potential applications/questions, and most of the applications had one of the following forms:

- Should I/we do x?
- How can I/we do x?
- Who/what/where/[which x] should I/we y?

So, most of the ideas are about whether one ought do something or how one ought to do it. In terms of topics, the most common ones were:

- **Health** (e.g. How can I sleep better? How can I lose weight?)
- **Entertainment** (e.g. What book should I read next? What should I do for fun today?)
- **Work and research** (e.g. How can I be more productive? What company should I work for?)
- **Life planning** (e.g. What career is right for me? When should I get married?)
- **Money** (e.g. How can I earn $1000 on the side? How should I prepare for retirement?)
- **Social** (e.g. How can I find a girl/boyfriend? How can I improve my long-distance relationship?)
- **General self-improvement** (e.g. Are there any big mistakes I am making? What can I improve in my life?)
- **Products** (e.g. What clothes should I wear? What mobile phone should I buy?)
- **Learning** (e.g. How can I learn Spanish? What skills should I learn?)
- **Analysis and explanation** (e.g. Why did Obama win the election? How do bicycles work?)

In the long run, I expect that many questions will come from the long tail and won't have been asked in their exact form before. However, in the beginning it is worth focusing on specific questions, so that answer strategies can be reused across conversations.

I made a quick attempt to score potential applications based on (a) how important the problem is for the audience, (b) how completely I expect that we can solve it, and (c) how plausible it is that we can monetize it (through direct payments, affiliate fees, etc). This may be worth doing more systematically, and with better criteria; my brief stab at it resulted in the following list of applications that scored well in terms of a multiplicative combination of the three features:

- Should I buy or rent?
- How can I immigrate to {country}?
- How can I be happier?
- How can I find a girl/boyfriend?
- How can I get laid?
- What startup/company should I work for?
- How can I make $1000?
- How can I gain/lose weight?
- How can I be healthier?

- How can we improve the success probability of our company?
- How can I sleep better tonight?
- Why do I feel anxious/depressed?
- Should I stay at my current job or change jobs?
- How can I get a raise?
- What doctor should I go to?
- How can I have more energy?
- (When) Should I/we get married?
- (When) Should I/we have children?

8.2.3 A first application—proposal #1

Let's consider the following question as a first application:

> **"How can I feel better right now?"**

For this question, I am imagining that the dialog would, in its initial phase, go through a mostly automated flow chart (or checklist) of issues, comparable to this online tool, or a more short-term version of one of these self-care checklists. Towards the end, it could turn into more of an open conversation with an expert, in some ways comparable to talk therapy tools like 7 Cups of Tea, Kindly, or BlahTherapy.

Comparison to existing tools In contrast to these existing tools:

1. We interpolate more smoothly between the two modes. Within a single dialog, the possibility to diverge from the more automated "flowchart" path is always there; the user can always provide a manual response as opposed to one of the provided multiple-choice responses. Across dialogs, we learn over time how to automate bigger parts of these sorts of conversations.

2. We put more emphasis on making things easy for the user. In particular, we only ask the user for information when we expect to need it and can't obtain it in another way, and also try to take multiple choice as far as possible.

3. We improve more over time, accumulating knowledge on how to have helpful conversations on this topic (in the form of "objective" sub-dialogs that human contributors take into account), and integrating that knowledge into algorithms (that learn from what human contributors do).

4. We do want our users to pay, e.g. in the form of tips after the dialog is over, as a requirement to continue past some point, or in the form of optional pledges that incentivize more contributors to join the conversation.

How good is this application? How does this application fare with respect to the criteria for good initial applications that I outlined earlier? I'd say the problem is definitely important to the target audience and people would want to use it more than once if it worked. It's very likely that there is a natural progression to other topics, and that the amount of information required from users is at the sweet spot between too little and too much. It's probably the case that simple automation can go a long way and that our solution will be much better than the next-best one. I'm less sure that the app wouldn't paint a misleading picture of the longer-term goals for dialog markets and that people would be willing to pay for helpful advice.

Potential issues A potential issue with this applications is that it is related to a wide range of other topics. This could prevent us from solving the problem to the extent that users expect us to solve it, at a price that users are willing to pay, and with the level of automation that I am imagining. I expect that the system will work fairly well in the beginning of dialogs (using decision trees of multiple choice questions as a baseline strategy), and that it can work fairly well towards the end of dialogs (recovering one-on-one conversation), but I have some uncertainty about the intermediate stage that requires the crowd to choose next steps, guided by "objective" dialogs that provide instructions on how to perform well, and an incentive system that strongly encourages contributions that follow these instructions.

A second potential issue is that users will expect near-real-time interaction for this application due to its short time horizon. For longer-term questions such as "What career should I pursue?", it is more acceptable for follow-ups to happen only occasionally, and with some delay in between. For this application, users probably expect quick back-and-forth. This is feasible using automation, but once the dialog switches to the mode where contributions come from the crowd, it may slow down. Depending on how much of a slowdown this is, and depending on how well we manage users' expectations, this could lead to a frustrating experience.

A third potential issue is that, for this particular application, some of the benefit to the asker might be derived not from the content of the conversation, but from the mere fact that they are having a conversation with someone (or some system) that cares and listens. If the activity per se is responsible for a big part of the benefit, this application would not be ideal, since it could be different in kind from follow-up applications where the goal is more directly to think through a question and provide relevant information.

Example of a sequence of follow-up applications The application above is related to many topics, including health, exercise, sleep, relationships, work, money, and life planning. This makes it difficult to say what a likely trajectory might be. Actual expansion plans will depend on empirical feedback, based on what sub-dialogs happen frequently and how well they score according to our desiderata for applications. Still, for the sake of concreteness, here is an example trajectory that includes a few potential next applications:

1. How can I feel better right now?
2. Should I seek professional help for mental health?
3. What kind of exercise should I do?
4. How can I make new friends?
5. How can I sleep better?
6. How can I improve my relationship?
7. How can I be healthier?
8. How can I be happier in general?
9. How can I earn more income?
10. How can I find purpose in my life?

11. What career is right for me?

8.2.4 A first application—proposal #2

Let's consider another potential first application:

> **"How can I find a girlfriend/boyfriend?"**

I'm more uncertain about how conversations would go for this application. I can imagine that contributors would want to think through ways to meet potential partners who might be a good fit, and how to increase the probability that such meetings will lead to a good relationship. To that end, the dialog might initially elicit some personal information, such as the age of the person asking, where they live, what they do for work/school, and what they are looking for in a partner. We might then try to figure out how we can help—for example, whether we should focus on finding better places to meet partners or whether we should focus on improving the asker's chances given such meetings. Finally, we might work towards coming up with actionable steps in those areas.

Comparison to existing tools The overall problem under discussion is that of moving a person from the state where they don't have a girlfriend/boyfriend to the state where they do. Different existing solutions address different parts of this problem. Dating sites and apps (such as OkCupid, Match.com, Tinder) help a person meet and approach potential partners online. MeetUp and Facebook Events help with finding opportunities to meet potential partners offline. Various informational websites (such as WikiHow), blogs, podcasts, and books help with advice on how to increase one's attractiveness and approach to dating in order to improve the chances that meetings will lead to a relationship. Forums and subreddits (such as r/dating_advice) may also help with that. There are also matchmakers (who help with finding dates), dating coaches (who help with appearance, conversation, etc.), and dating seminars.

Compared to dating sites and apps, MeetUp and Facebook, we wouldn't try to organize the interaction of multiple people people online, but rather focus on other parts of the problem. Compared to existing informational sites, blogs, podcasts, and books, our system would be much more interactive and we would try to do as much of the work as possible for users. We would require less initiative from users, and users wouldn't need to start out with an accurate view of their situation for our approach to succeed. Compared to forums and subreddits, we would have quicker back-and-forth, better incentives for helpers, and we would improve more over time. Compared to matchmakers and dating coaches, our system would be cheaper, more anonymous, and there would be less friction in getting started.

How good is this application? How does this application fare along the criteria I outlined earlier? It's definitely the case that the problem is important to the target audience and that there is a natural progression to other topics. I expect that the app doesn't paint a misleading picture of the longer-term goals for dialog markets. I'm not sure whether the amount of information required from users is at the sweet stop between too little and too much, whether simple automation can go a long way, whether our solution will be much better than others, and whether people would be willing to pay. By its nature, people wouldn't use the system more than once if it worked very well.

Potential issues My main uncertainty is about how far simple automation can take us, and about the consequences thereof for cost and quality of our solution. For some other applications (such as "How can I feel better now?" and "Should I buy or rent?"), there are existing automated systems that are useful for at least some people. For this app, I don't know of such existing systems. I also expect that the dialogs for this app will be substantially longer than for more self-delimited questions such as "Should I buy or rent?". Whether this creates a problem depends on how much custom labor is required per dialog. It is easy to imagine that only a small fraction of these dialogs can be automated, even with plenty of training data. This will be the case if each asker eventually requires highly customized coaching, as opposed to relatively generic advice chosen from a limited set and adapted to the asker's circumstances. For example, this could happen if askers find it difficult to accept advice, so that most of the dialog is not about figuring out what advice to give, but rather about how to communicate it in a way that is convincing to them, and if there is a lot of variation in what is convincing to different people.

There are also a variety of potential social/privacy issues associated with this application. For example, to most effectively help, it could be useful to know what the asker looks like, but this makes the asker more easily identifiable, which could raise privacy concerns given that the asker's information is accessible in our semi-public (NDA-gated) market. This application could also attract contributors who are not primarily motivated to help, but rather are seeking dates, participating for entertainment value, trolling, etc. The privacy issue can be addressed to some extent by marking particular kinds of content as access-restricted, so that only a limited subset of trusted users can see it, but this seems like a complication that it would be best to avoid initially.

Example of a sequence of follow-up applications As above, I expect that the actual series of follow-up applications will depend on what sub-questions occur frequently, and what nearby questions these sub-questions suggest. Still, here is an example of how things could go:

1. How can I find a girlfriend/boyfriend?
2. How can I be more attractive to potential partners?
3. Where can I meet potential partners?
4. How can I make friends?
5. What should we do for our first date?
6. What activity should I do with my significant other?
7. What activity should I do with my friends?
8. How can I improve my relationship with my significant other?
9. How can I improve my friendships?
10. Should I stay in my current relationship?
11. (When) Should we get married?
12. (When) Should we have children?

8.3 Where do we go from here?

At this point, we could think through more potential first applications, or we could think through the aforementioned ones in more detail. However, it may be more useful to gather empirical feedback. I suggest the following procedure:

1. Pick one of the applications mentioned above; ideally one where you have some expertise, or can gather some expertise within a relatively short amount of time.

2. Create a website that automatically starts a chat between yourself and any given visitor. Make it easy for yourself to send multiple-choice questions to the visitor. Perhaps style the website so that it looks a bit less like a chat and a bit more like a multiple-choice quiz to get over users' potential aversion to talking to strangers.

3. To acquire users, run a few ads on Google Adwords or Bing for searches that correspond to the application.

4. Gather dialogs until you get a sense for what they tend to be like. Then repeat with another candidate application.

I expect that this will be instructive. It will help you understand how much repetition there is between users, how willing people are to keep up such dialogs, and how much information needs to be elicited to provide helpful advice. As an additional step, you can try different payment schemes. Would people be willing to tip after the dialog is over? Would people be willing to pay to continue the dialog after some fixed number of exchanges?

The user experience won't be identical to the true dialog market setting. The responses will be less well-informed, but perhaps more coherent, since they come from a single person. In the simple version outlined above, users won't be able to access a tree-structured dialog that reflects the current state of problem solving. Nonetheless, the simulation of the user experience is close enough that you should be able to gain a fair amount of certainty that an application is viable without implementing the real dialog market just yet.

Acknowledgments

This project has benefitted from discussion with and feedback from many people, more than I can name here. For particularly helpful and in-depth advice, I thank Paul Christiano, Owain Evans, and Noah Goodman.

I am also grateful to Leon Bergen, Jamie Brandon, Frederick Callaway, Cameron Freer, Roger Grosse, Robert Hawkins, Frauke Harms, Daniel Hawthorne, Sha Jin, Justine Kao, Benedict Küster, Falk Lieder, John McCoy, Jon Malmaud, Long Ouyang, Desmond Ong, Alexey Radul, John Salvatier, Daniel Selsam, Natalie Schaworonkow, Alexandra Surdina, Michael-Henry Tessler, Tomer Ullman, Alyssa Vance, Mike Webb, Jess Whittlestone, and Jeff Wu.