CloseUp—A Community-Driven Live Online Search Engine

CHRISTIAN VON DER WETH, ASHRAF ABDUL, ABHINAV R. KASHYAP, and MOHAN S. KANKANHALLI, National University of Singapore

Search engines are still the most common way of finding information on the Web. However, they are largely unable to provide satisfactory answers to time- and location-specific queries. Such queries can best and often only be answered by humans that are currently on-site. Although online platforms for community question answering are very popular, very few exceptions consider the notion of users' current physical locations. In this article, we present CloseUp, our prototype for the seamless integration of community-driven live search into a Google-like search experience. Our efforts focus on overcoming the defining differences between traditional Web search and community question answering, namely the formulation of search requests (keyword-based queries vs. well-formed questions) and the expected response times (milliseconds vs. minutes/hours). To this end, the system features a deep learning pipeline to analyze submitted queries and translate relevant queries into questions. Searching users can submit suggested questions to a community of mobile users. CloseUp provides a stand-alone mobile application for submitting, browsing, and replying to questions. Replies from mobile users are presented as live results in the search interface. Using a field study, we evaluated the feasibility and practicability of our approach.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing**; • **Information systems** → *Web searching and information discovery*; *Crowdsourcing*;

Additional Key Words and Phrases: Live online search, community question answering, crowdsourcing, social computing, collaborative service, query transformation

ACM Reference format:
Christian von der Weth, Ashraf Abdul, Abhinav R. Kashyap, and Mohan S. Kankanhalli. 2019. CloseUp—A Community-Driven Live Online Search Engine. *ACM Trans. Internet Technol.* 19, 3, Article 39 (August 2019), 21 pages.
https://doi.org/10.1145/3301442

1 INTRODUCTION

Search engine (SE) giants such as Google or Bing seem all powerful when it comes to finding information online. Once information is on a Web site, it gets quickly indexed by SEs and thus is easy to find. However, traditional SEs suffer from a major blind spot. Imagine a user searching for a restaurant to have dinner or for a movie to watch in a cinema. Answers to the following questions could be very valuable: "Is there currently a long queue outside *Slurpilicious*?" or "Is the

This research was supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Strategic Capability Research Centres Funding Initiative.
Authors' address: C. von der Weth, A. Abdul, A. R. Kashyap, and M. S. Kankanhalli, School of Computing, National University of Singapore, I3 Building, #02-02, 21 Heng Mui Keng Terrace, Singapore, 119613, Singapore; emails: {chris, aabdul, abhinav, mohan}@comp.nus.edu.sg.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.
1533-5399/2019/08-ART39 $15.00
https://doi.org/10.1145/3301442

screening of *Deadpool 2* at Cinemax very crowded?" These and similar questions have in common that they (maybe implicitly) reference a specific time and location. As such, valid answers typically expire too quickly and are only of interest for a too limited number of users to be maintained on a Web site, let alone being indexed by SEs. However, people who are physically in or close to the restaurant or cinema could easily provide answers.

Reaching out to other users to get information has always been an important facet of the Internet, with Bulletin Boards (1978), Internet Relay Chat (1988), and even the first forms of social networks predating the World Wide Web (WWW) as we know it today. The advent of the WWW in 1989 and particularly the so-called Web 2.0 around 2004 spurred the success and proliferation of community-driven platforms such as online forums and question answering sites (e.g., Quora, Yahoo! Answers). Other services for information seeking that rely on or heavily benefit from user-generated content are recommendation sites (e.g., TripAdvisor, Yelp), social news sites (e.g., Reddit, VOAT), and social bookmarking sites (e.g., Diigo, Pearltrees). Such communities thrive on the absence of physical boundaries on the Web but make them unsuitable for finding live and on-the-spot information.

The advances in mobile technologies and the prevalent "Always On" lifestyle of users allow for novel ways to engage the wisdom of the crowd. Services that rely on data stemming from users' mobile devices or users' explicit input are gaining increasing popularity. For example, the navigation app Waze automatically collects travel times and other journey information to estimate road and traffic conditions for route planning. Users can also report accidents, traffic jams, and so forth. Google Maps collects anonymized location data from its user base to estimate how crowded venues such as restaurants or bars currently are. Google Maps also introduced a Q&A service asking visitors of a venue, for example, whether the place is good for groups and if it has outdoor seating. The focus is not live search, however, but to build up a knowledge repository to improve traditional Web search.

As the next evolutionary step, we propose the integration of traditional Web search and community question answering (CQA) to support the search for live and on-the-spot information. With Web search powered by machines and CQA powered by humans, there is a wide gap to bridge to fuse both forms of information seeking. To this end, we present CloseUp, our prototype to facilitate the seamless integration of CQA into Web users' online search experience. CloseUp extends a Google-like search interface to suggest and submit relevant live questions about points of interests (POIs), as well as presents incoming answers from mobile users as search results. The heart of CloseUp is a query-to-question (Q2Q) neural network pipeline that analyzes and translates keyword-based queries into well-formed questions. Each question is sent to users having the CloseUp mobile app installed and are close to the identified POI. Apart from evaluating the accuracy of the Q2Q pipeline, we conducted a field study where participants used our mobile app to answer live questions for 2+ weeks. More specifically, we make the following contributions:

(1) Outline of challenges. Given the novelty of our approach for the integration of Web search and CQA, we first motivate fundamental challenges toward this goal. This particularly includes the formulation of search requests (keyword-based queries vs. well-formed questions) and the expected response times (milliseconds vs. minutes/hours), but also incentives and user loyalty.

(2) CloseUp prototype. We present the overall architecture of CloseUp—first, its two frontend applications: a Web search interface for the support of submitting questions, as well as displaying new incoming answers over time, and a mobile application to submit, browse, and reply to questions. We then give an overview of the core backend components for query and question processing.

(3) Q2Q pipeline. The Q2Q pipeline to convert keyword-based queries into questions comprises three steps: (1) the identification of relevant queries, such as queries referring to live information

about POIs, (2) the extraction of POI names from queries, and (3) question suggestion, such as the automated translation of queries into well-formed questions. To this end, we adopt state-of-the-art deep learning models for named entity recognition (NER) and sequence-to-sequence translation.

(4) Training datasets. Due to the lack of appropriate datasets, we propose a data-driven generation of synthetic datasets of annotated queries and questions to train our Q2Q pipeline. We generate questions based on users' interests expressed in TripAdvisor reviews. From these questions, we derive search queries that comply with common query patterns. All datasets are publicly available.

(5) Evaluation. We evaluate the accuracy of the Q2Q pipeline using established measures as well as crowdsourcing experiments. We also conducted two user experiments. In a field study, participants used the mobile app for 2+ weeks and replied to location-dependent questions about POIs. In a lab study, participants evaluated the Web search interface. We analyze the resulting data and the survey where we asked participants to share their experiences with CloseUp.

The article is organized as follows. Section 2 outlines the core challenges toward live online search. Section 3 reviews related work. Section 4 presents the main components of our current prototype. Section 5 details on the Q2Q pipeline for the classification and translation of search queries. Section 6 describes the process of generating our synthetic datasets. Section 7 presents the results of our evaluation. Section 8 discusses our ongoing and future research directions. Section 9 concludes the article.

2 CHALLENGES FOR LIVE ONLINE SEARCH

Web search and CQA have been integrated to the extent that the content of CQA platforms is indexed by SEs. This assumes, however, that the information is available in form of forum posts or submitted answers. Yet this assumption no longer holds for the latest information about physical places (see the examples in Section 1). In the following, we outline the challenges to make live information that is not available on Web pages searchable in a Google-like manner:

(1) Queries versus questions. When users submit questions to online forums or Q&A sites, they do so using proper sentences. This reduces the risk of ambiguities and is de facto the only accepted mode of written communication in online communities [64]. When it comes to online search, however, most users formulate keyword-based queries, often underspecified, ambiguous, and multifaceted [4, 70]. Modern SEs apply a variety of query expansion, query suggestion, and query refinement techniques to implicitly or explicitly address these issues to support users in their search [7, 48]. Similar efforts are needed to close the gap between keyword-based search queries and well-formed questions to be asked in a community forum.

(2) Delayed responses. The results from SEs are typically expected to arrive within milliseconds. In contrast, users accept that useful answers to questions posted in online forums or on Q&A sites can take minutes, hours, or even days [63]. With CloseUp, we currently focus on live questions that require answers in less than 1 hour, preferably within only a few minutes. This calls for changes to the interfaces of the SE to accommodate new incoming results over time. But more importantly, it requires a changing mind-set when it comes to Web search, with users no longer able to expect the best results immediately after submitting a query.

(3) Incentives. Most online communities reportedly suffer from undercontribution [53]. The often observed "90-9-1 Principle" states that only 1% of the population of a community actively contributes on a regular basis, and 90% passively consumes content; the 9% refers to the occasionally active users. Traditional online forms and Q&A sites can still flourish considering that 1% to 10% of the population is still a sufficiently large number of users. In contrast, the number of people who can potentially answer questions like "Is there a long queue outside Restaurant X right now?" is inherently very limited. In general, users are willing to contribute if their perceived benefits

outweigh their perceived costs. Considering that benefits and costs depend on many and often less tangible factors [34], at this early stage, we focus on minimizing the cost for users mainly in terms of the effort to answer questions.

(4) User loyalty. Simply speaking, the backend clusters of Google or Bing do not complain about how many search queries they have to process or how meaningless many queries there are. Sending users too many and irrelevant unsolicited questions, however, can have severe negative consequences. Users might feel harassed, exploited, or put off by low quality of the provided service(s), all very common reasons why users stop contributing or even leave online communities [5]. Although closely related to creating incentives, it also addresses the challenge of not driving away even the most motivated and engaged users.

Summing up, there are two main reasons why community-driven live online search is challenging: search results do not come from machines but human users, and real-world CQA platforms inherently impose some form of physical boundaries on a community. In this article, we address these challenges and show in our evaluation the practicality of our approach. We consider this work as the first stepping stone toward bringing together classic Web search and CQA for seeking live information in and about the physical world.

3 RELATED WORK

Community Question Answering. Q&A sites like Quora, Yahoo! Answers, and Wiki Answers allow users to ask complex, multifaceted questions that allow for different good answers but also facilitate valuable discussions over questions and answers [1]. Srba and Bielikova [60] present a comprehensive classification of CQA platforms. Considering that answering questions involves effort for users with no direct payoff, CQA represent a so-called *information-exchange dilemma* [33] with the risk of such sites suffering from undercontribution. Various works investigated what motivates users to invest time and effort to answer questions, highlighting users' satisfaction, knowledge self-efficacy, self-presentation, and peer recognition as the main motivators [28, 29]. Other works also evaluated the positive effect of more extrinsic incentives such as reputation and gamification systems (reputation scores, badges, privileges, etc.) [16, 25, 32]. Question routing aims to identify the users best suited to answer based on their profile or expertise [9, 11, 55]. A lot of work has been done to make CQA knowledge searchable from an information retrieval perspective [59, 73]. The main challenge is the lexical gap, given that two differently worded questions can ask for the same information. However, existing Q&A sites are unsuitable for asking time- and location-specific questions, where users' "expertise" mostly depends on their current locations.

NER in Queries. NER is the well-established natural language processing (NLP) task of identifying the names of persons, organizations, locations, and so forth [46]. Conventional NER is performed over well-formed text documents and as such can rely on proper spelling, grammar, capitalization, and punctuation [52]. With the rise of social media, NER has been adopted to handle (more) informal text. Given the availability of the data, most works applied NER on tweets [37, 38, 41]. NER has also been applied on search queries, with keyword-based queries typically not adhering to any proper grammar. One of the first works done by Microsoft Research uses a weakly supervised latent Dirichlet allocation to identify the names of movies, games, books and songs in queries [23]. Yahoo! Research proposed a two-step process using conditional random fields (CRFs) to first identify named entities in queries and then assign them to one out of 29 predefined categories [19]. A CRF-based approach is also proposed by Expedia to find named entities in travel-related search queries [14].

Sequence-to-Sequence Learning. The Q2Q translator of CloseUp uses a sequence-to-sequence recurrent neural network (RNN) architecture. Such architectures consist of an encoder that encodes an input sequence and a decoder that decodes an output sequence conditioned on the input.

Simple sequence-to-sequence models [12, 61] and models that employ weighted attention over the input sequence [3, 40] have significantly improved machine translation tasks. These models and their enhancements have also significantly improved human-like abstractive summarization. For example, attention-based models have been proposed to perform abstractive single-sentence summarization that perform paraphrasing, generalizing, and compressing the original sentence [13]. For multisentence abstractive summaries, Nallapati et al. [47] employ copying mechanisms to deal with words not appearing in the training dataset for generating summaries. Sequence-to-sequence models have also been applied to conversational modeling [66] and speech recognition [8]. In general, these models have shown to be effective for many tasks, and we employ such a neural network architecture to transform keyword-based queries into well-formed questions.

Crowdsourcing Platforms. Utilizing the wisdom of the crowd in an organized and structured manner has been pioneered by crowdsourcing platforms [17] such as Amazon Mechanical Turk. Such platforms allow individuals or organizations to submit microtasks—typically very difficult for computers but easy for humans like image labeling or sentiment analysis—that are then performed by a workforce of volunteers for a small payment; several excellent surveys [26, 76, 77] about crowdsourcing platforms are available. With the advances in mobile technologies, platforms for mobile crowdsourcing where tasks depend on users' locations gained a lot of traction [57]. Common tasks include price and product placement checks in brick-and-mortar stores, location-aware surveys and data collection, property and product verification, and so forth [45, 54]. This demand spurred the development of many research prototypes [31, 65, 74]. Mobile crowdsourcing platforms can vary significantly, making a meaningful classification challenging [2]. Although we do not exclude monetary incentives in principle, we consider CloseUp a community-driven platform akin existing Q&A sites and online forums. Introducing money can have unexpected effects on social norms, particularly when introducing micropayments [21], which are beyond our current scope. Closest to our approach is MoboQ [39], which provides a mobile app allowing users to ask and reply to time- and location-specific questions. In contrast, the main contribution of CloseUp is the seamless integration of live questions and answers into a Google-like search experience, with the mobile app representing one component of the system. We therefore focus less on app-specific challenges but on the required solutions to bridge the gap between traditional Web search and CQA.

4 THE CLOSEUP PLATFORM

Figure 1 shows the current architecture of CloseUp with screenshots of the Web search interface and mobile app, as well as the core backend components. We provide more information, screenshots, video clips, all datasets, and the link to the Web search interface on our project Web site.[1]

4.1 Frontend Applications

Given the two roles of searching online users and mobile users, CloseUp provides two frontend applications—a mobile app and a Web search interface—to accommodate both user roles.

Mobile App. The CloseUp mobile app is a stand-alone application to submit, browse, and reply to questions.[2] In CloseUp, each question is associated with a geocoordinate and a time to live (TTL) in minutes reflecting at what point answers are no longer considered valid. Users submit *public questions* by first selecting a location on a map. For each question, users need to specify a TTL and a set of up to five possible answers. Public questions are displayed on the map as markers visible for all users. To reply to a question, users click on its marker and select the appropriate answer in the

[1] http://185.170.113.47/closeup/.
[2] The app is currently available for Android at https://play.google.com/store/apps/details?id=sg.edu.nus.closeup.

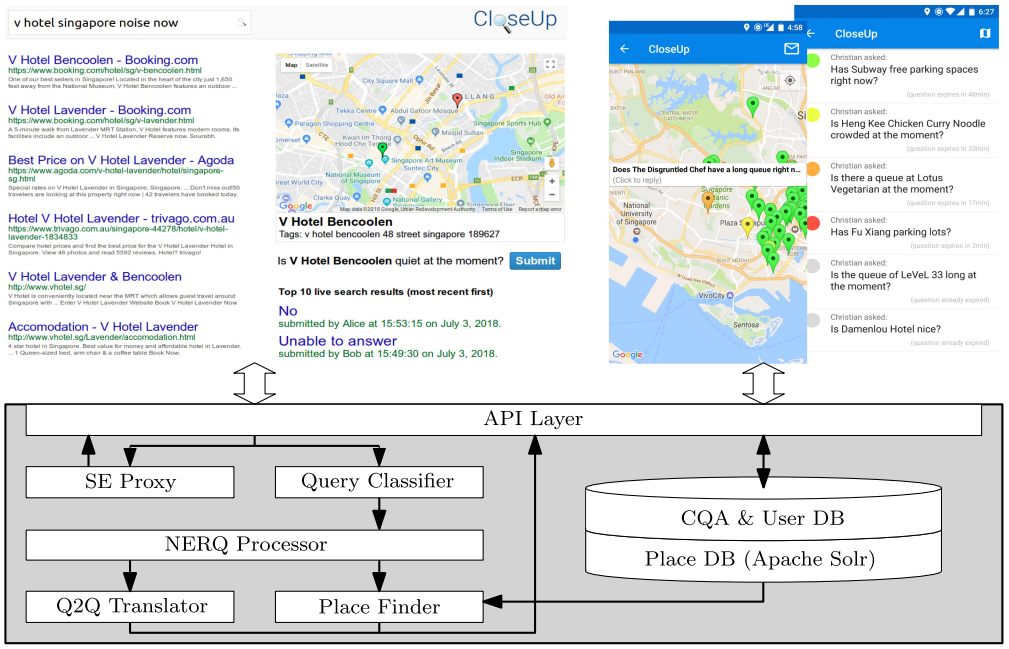


Fig. 1. System architecture of CloseUp.

displayed dialog. A list view shows a user's submitted questions and received answers. Users can also receive *private questions* in an email-like in-box and reply to them. Each new private question triggers a notification on users' devices. The application keeps track of users' current locations and activity states (still, walking, in vehicle, etc). The intuition is that users are not considered available for answering location-specific question if they are driving in a car or taking public transport. Any changes in users' locations and activity states are sent to the backend server, including when users switch on the locations service on their device (e.g., to save battery life). Again, the rationale is to determine the state when users are not available to reply to questions.

Web Search Interface. The Web interface imitates a Google-like search experience and also shows the top 10 search results from Google for each submitted query. If a submitted query is classified as a CQA query, the search interface suggests a relevant question and a set of relevant POIs as markers on a map component. Users can click on markers to select a POI that in turn updates the suggested question. Last, users can submit questions to the backend server. When loaded in the browser, the Web interface opens a new WebSocket connection to the CloseUp server. The server uses this connection to send new answers back to the client browser. The connection is open during the whole search session. Each new answer is displayed on the site as a new live search result (beside the Google results). A single search session may comprise of an arbitrary number of submitted queries and questions. Although the Google results get updated for each query, the live search results only depend on the stream of incoming answers. Besides the answer, each live search result is also accompanied by the original questions to distinguish between the different submitted questions.

4.2 Backend Architecture

The two frontend applications communicate with the backend using a representational state transfer (REST) application programming interface (API). Each query is first passed to the SE Proxy and the Query Classifier (QC). The SE Proxy uses the query "as is" to request the search results from

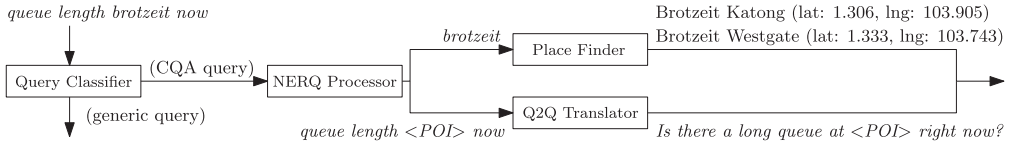


Fig. 2. Example of a submitted search query passing through the Q2Q pipeline.

a commercial SE (currently Google) and passes the results back to the Web interface. For the time being, the QC is a binary classifier that decides if a query is a *generic query* or a *CQA query*. Each CQA query is then passed to the NERQ in queries (NERQ) processor, which identifies the name of the POI. We currently assume that a CQA query contains only one POI. The output of the NERQ processor is used as input for the Q2Q translator and the Place Finder. The Q2Q translator converts the query into a corresponding well-formed question. The Place Finder uses the extracted POI name to query a place database. The results of both components are returned to the Web interface.

Questions submitted via the Web search interface are treated as public questions and as such are visible on the map of the mobile app. If a mobile user replies to such a question, the answer is forwarded to the Web interface via the established WebSocket connection. Depending on mobile users' current locations, the backend also sends a submitted question to users' personal in-box in form of private questions: if a mobile user is within, say, 200m of a submitted question, that user will get this question pushed to his or her in-box and notified accordingly. Finding the optimal value for the radius—which arguably should be dynamic and depend on users' current locations, the number of available users around the point of interest, and so forth—is beyond the scope of this article. To address the often reduced accuracy of GPS, particularly indoors, we use a sliding window over users' most recent locations and apply convex hull peeling [18] and simple averaging to better estimate a mobile user's true location. A repository stores all user-related information, as well as all public and private questions, together with submitted answers. Note that public questions can feature multiple answers from different users.

5 Q2Q PIPELINE

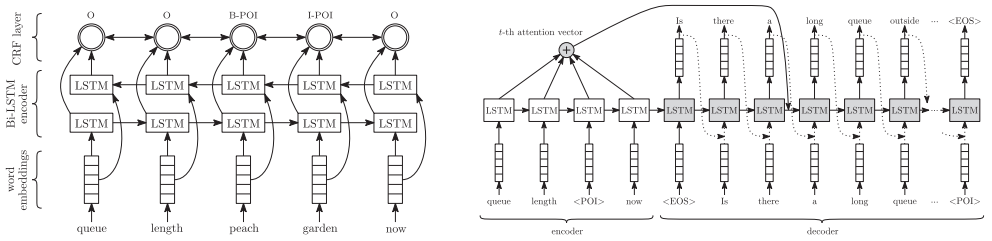
For the Q2Q pipeline to translate queries into questions, we adopt state-of-the-art deep learning models. Figure 2 shows an example for a query passing through the pipeline.

5.1 SE Proxy

The goal of the CloseUp Web search interface is to provide a Google-like search experience. We therefore not only display CloseUp-related information and results but also native search results from commercial SEs. Note that not all queries are necessarily CQA queries. We currently retrieve results from Google, but the SE Proxy can be extended to include other SEs.

5.2 Query Classifier

We use FastText [30] to decide whether a query is a generic or a CQA query. Although we pass both query types to the SE Proxy, we only pass CQA queries to the Q2Q pipeline. This minimizes the number of processed queries (efficiency) but also minimizes the number of arguably less meaningful questions (effectiveness). For example, the query "nice soup spoon" is more likely to be a generic query with the intent to search for and buy soup spoons. However, the NERQ processor might label "soup spoon" as the place name—"The Soup Spoon" is a local restaurant chain—which in turn might yield the question "Is Soup Spoon nice?" Although being a valid question, it does



(a) NERQ network model: Bi-LSTM-CRF Conditional Random Field (CRF) on top of a bidirectional Long Short-Term Memory (LSTM) network

(b) Q2Q translation network model: LSTM-based encoder and decoder with attention (only one attention vector connection shown to ease presentation)

Fig. 3. Network models of the NERQ and Q2Q classifiers.

(more likely) not match the intent of the query. In the long run, we envision to support more types of queries, such as when search results can best be provided by physical sensors [36].

5.3 NERQ Processor

We implement a bidirectional–long short-term memory (LSTM)-CRF (bi-LSTM-CRF) network model that stacks a CRF on top of a bidirectional LSTM network [35] (Figure 3(a)). For the training and evaluation, we set the size of the word embedding vectors to 128 and the size of the hidden layer to 512. The model takes a query as input and outputs a sequence of labels in the common inside-outside-beginning (IOB) tagging format. Considering that we focus on place names, we only rely on `B-POI` and `I-POI` to mark the beginning and inside of POI names, and Outside `O` to mark all other terms in the query. Using the sequence of IOB labels for a given query q, we convert q to a new query q' by replacing the identified place name with the special token `<POI>`. The rationale is that the words in place names are often rare—this is particularly true for restaurant names with a wide range of international cuisines—and that place names themselves do not need to be translated [22]. For example, for a query q = "queue length peach garden now" and label sequence `[O,O,B-POI,I-POI,O]`, we yield a new query q' = "queue length <POI> now." Query q' serves as input for the Q2Q translator, whereas we use the POI name as search query for the Place Finder.

5.4 Place Finder

The NERQ classifier only returns the phrases that represent the names of POIs. Particularly given the large number restaurant chains and franchises, extracted names are often ambiguous. To resolve this, we present users with the top-n most likely places in the map section of the Web search interface. To this end, we submit an extracted POI name as query to the Place Finder, a component based on Apache Solr to index and search for information about places. As a dataset, we used the Google Places API to collect the information about 177k+ places within Singapore. Note that we index not only the names of places but also a set of tags derived from the name and any additional information, mainly the address field. This often enables to resolve ambiguities, as outlets or branches of the same chain or franchise are often identified by, for example, the street name. Given a query, the Place Finder returns a ranked list of places with their names, sets of tags, and geocoordinates.

5.5 Q2Q Translator

For the Q2Q translator, we adopt a sequence-to-sequence learning model [61] with an RNN-based encoder and attention decoder [40] (see Figure 3(b)). Both encoder and decoder have been trained with two hidden layers of LSTM units of size 256. The attention decoder applies *teacher forcing* [71]

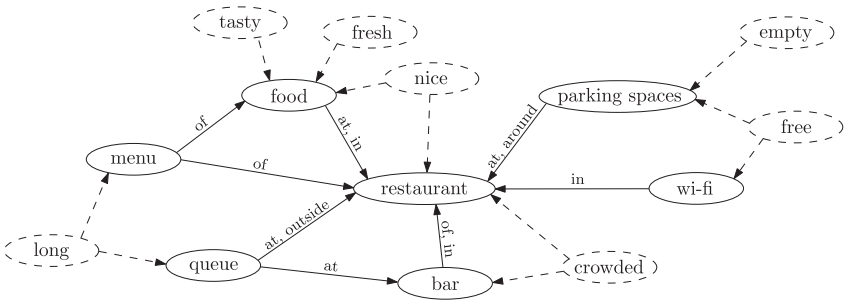


Fig. 4. Example of a generated word graph. Solid nodes represent nouns, and dashed nodes represent adjectives. Dashed edges reflect commonly used adjective-noun pairs. Solid edges labeled with a set of prepositions reflect commonly used noun-preposition-noun-pairs.

with a rate of 50% and dropout with a rate of 5%. Each data point for the training is a query-question pair with place names represented by the special token <POI> in both query and question. In the final pipeline, this replacement is done by the NERQ processor. All queries and questions are lowercase, and we omit the question mark at the end of questions. The output of the Q2Q translator is a sequence of words representing the predicted question. For example, for query "query length <POI> now," the predicted sequence might be "is there a long queue at <POI> right now." This sequence together with the ranked list of places as output of the Place Finder is returned to the Web search interface. The final question suggested to the user is generated on the client side by replacing <POI> with the respective name—by default, the name of the highest-ranked POI or the name of the POI selected on the map—and forming a proper sentence: capitalizing the first word of the sequence and adding a question mark (e.g., "Is there a long queue at Peach Garden right now?").

6 DATASET GENERATION

Although SEs or Web sites with a search function (e.g., travel and booking sites) have query logs, they do not share them due to privacy concerns. We further do not expect such query logs to contain many live queries, if any, given that existing platforms do no support this form of information seeking. Due to the lack of available real-world datasets of sufficient size required for training our Q2Q pipeline, we therefore propose a data-driven approach to generate synthetic datasets [20, 27] based on TripAdvisor reviews. In more detail, our dataset contains 215k+ reviews about 277 hotels and 193k+ reviews about 3,486 restaurants in Singapore (available for download on our project Web site). Our underlying assumption is that information shared in reviews is also the information that users are most interested in having when searching for a venue.

6.1 Word Graph Construction

Based on our review corpus, we create a word graph which allows us to generate well-formed questions. We extract two concepts from the corpus that form the basic building blocks of the graph: adjective-noun pairs (ANPs) and noun-preposition-noun pairs (NPNPs). Figure 4 shows an example. We used the Stanford dependency parser [10] for this task. Focusing on Yes/No questions, we only rely on forms of *to be* and *to have* as verbs and thus do not require a representation of verbs.

Automated Word Extraction and Linking. We extracted all ANPs to identify commonly used attributes to describe an aspect. For example, frequently occurring ANPs for restaurants are *long-queue*, *tasty-food*, or *crowded-restaurant*. ANPs allow for very basic questions such as "Is Restaurant

X crowded?" Although they also allow for questions like "Is the queue long?," these questions lack the specification regarding the place of interest. We then extracted all frequently used NPNPs such as *wifi-in-rooms* and *rooms-of-hotel*. This allows for the formulation of questions like "Is there free wifi in the rooms of Hotel X?" Although one can form arbitrary long chains of NPNPs, resulting sentences structures are very uncommon in the English language. We therefore never connect more than two NPNPs. Generating well-formed questions requires correct verb forms and articles. With our focus on live search, we currently limit ourselves to questions in present tense. Thus, verb forms and articles mainly depend on whether a noun is singular or plural and/or countable or uncountable. Additionally, we need to know if a noun refers to an abstract or physical entity, since abstract nouns typically require an adjective to result in meaningful sentences. For example, "Does the restaurant have nice atmosphere?" requires the adjective *nice* to form a meaningful question. We use existing toolboxes such as WordNet [43] and Pattern [15] but also online dictionaries to automatically extract these features for each noun.

Manual Annotations. In principle, the automatically generated word graph allows for the generation of meaningful questions. To further improve the results, however, we semantically enriched the graph by manually adding information to nouns and adjectives. To make the task tractable, we limited the word graph to the top 100 most mentioned aspect (i.e., nouns) for each category: hotels and restaurants. This reduced the overall set of nouns to 170 and the set of ANPs to 3.9k pairs; NPNPs do not require further annotations. First, we enriched the nouns with synonyms and popular alternative spellings (e.g., *wifi*, *wi-fi*). Second, we labeled ANPs depending on whether the answer to a respective question changes on average on a yearly, monthly, weekly, daily, hourly, or minute-by-minute basis. For example, the answer to "Is Restaurant X crowded?" is likely to change more frequently compared to "Is Restaurant X expensive?" This allows to add time-related phrases to questionssuch as "Is Restaurant X crowded at the moment?" And last, we enriched ANPs with nouns related to the adjective to reflect that most online search queries are keyword based, containing mostly (proper) nouns [4]. For example, if a user wants to know if there is a long queue at Restaurant X, a search query is more likely to look like "restaurant X queue length now" than "restaurant X queue long now." Note that a meaningful mapping from an adjective to a related noun does not depend on the adjective itself but on an ANP since adjectives can have different semantics depending on the context. For example, whereas a *long queue* can be associated with *length* or *size*, a *long check-in* rather refers to *duration* or *time*.

6.2 Question Generation

Using our annotated word graph, we generate questions for randomly selected aspects (e.g., *parking lots*) and randomly selected categories (e.g., *restaurant*). We consider three different types of Yes/No questions, one asking for an attribute of an aspect and two types asking for the existence of an aspect, such as the following:

- "Are the parking lots around Restaurant X empty right now?"
- "Are there empty parking lots around Restaurant X right now?"
- "Does Restaurant X have empty parking lots right now?"

Note that the questions are often not semantically equivalent and do not have to be for our training. Strictly speaking, the first question asks if *all* parking lots are currently empty. In case of *existence* questions, we always add an adjective to aspects to avoid trivial questions like "Does Hotel X have rooms?" Last, note that the process does not ensure the most formal writing style. For example, "Does Restaurant X have a long queue right now?" is arguably of inferior style than "Is there a long queue in front of Restaurant X right now?" Language is often very subtle even in the context of simple sentences, and taking every nuance into account is beyond the scope of this article.

6.3 Query Generation

The input for the NERQ and Q2Q tasks are queries. In principle, users can submit proper questions as search queries, and we reflect this by mapping a subset of questions onto itself and consider it as a set of queries. For all other questions, we perform a series of steps to convert them into queries to reflect the commonly observed nature of search queries [4, 70].

Basic Keyword Extraction and Conversion. We first lowercase all terms and remove all stop words, including prepositions and all forms of *to be* and *to have*. We then randomly decide to convert adjectives to related query nouns—given by our manual annotation. We also place the adjective (or related query noun) and noun of ANPs adjacent to each other. Thus, for the question "Is the queue at Restaurant X long right now?," we might yield the query "queue line restaurant x now."

Keyword Modification and Extension. Given the simple structure of Yes/No questions and the limited vocabulary—apart from place names—basic queries were often "too clean" and lacked a "natural diversity." To make the classifiers more robust and allow them to better generalize, we augmented the data [62, 72] by adding noise in terms of additional words to the a query. We used the Word2Vec [42] model to learn word embeddings using a corpus of English Wikipedia articles as input. Using these embeddings, we then identified for each keyword in the basic query (excluding names of POIs) the 10 most similar words. From this set, we randomly selected zero to three words and added them to the query (e.g., resulting in "length waiting line restaurant x now wait span").

Keyword Reordering. Considering that keyword-based queries do not exhibit any notable grammar, we consider the queries "length waiting line restaurant x wait span" and "restaurant x wait now waiting line span length" as equivalent and equally likely. We reflect this by reordering query keywords in a semirandomly manner: we do not split and reorder (proper) nouns containing multiple words; we also we do not split ANPs but only change their internal order.

6.4 Generated Datasets

For the training, we generated two datasets to match the required input and output of the networks. The NERQ dataset contains search queries annotated using the standard IOB format to mark the beginning (`B-POI`) and inside (`I-POI`) of POIs; all other terms are labeled with `O`. The first half of the dataset are queries we generated using our word graph. To also include queries that do not contain POIs, the second half of the dataset are random queries with 2+ keywords from an AOL (formerly called *America Online*) query log [50]. We only removed queries where all terms are in the vocabulary of the generated queries. For example, we removed "nice soup spoon" since *Soup Spoon* is a popular local restaurant chain and "nice" is a commonly used adjective. All terms of the AOL queries are labeled with `O`. Note that we use the NERQ dataset not only for identifying location names but also to determine whether a query is a CQA query in the first place (see Section 5.2). The Q2Q dataset contains the query-question pairs. In all questions and queries, we replaced place names with a special token (`<POI>`) and omit the question mark at the end of questions. Both datasets are publicly available for download.[3]

7 EVALUATION

Our evaluation is divided into three parts. We first analyze our generated datasets and quantify the accuracy of the classifier components of the Q2Q pipeline. To evaluate the potential benefits and gain first insights into users' experience with CloseUp, we set up two user experiments: a long-term field study where participants used the mobile app and a lab study for the Web search interface.

[3] http://185.170.113.47/closeup/datasets/.

Table 1. Vocabulary Sizes of NERQ and Q2Q Training Datasets

	No. of Words	No. of POI Words	No. of Non-POI Words
NERQ Dataset	23,951	3,687	22,456
Q2Q Dataset	4,295	1*	4,294

*Only the special token <POI>.

Table 2. Recall, Precision, and F1 Score for QC and NERQ

	Recall	Precision	F1 Score
QC	99.8%	99.8%	99.8%
NERQ	98.6%	98.7%	98.7%

7.1 Q2Q Pipeline

We report the results of the three machine-learning components of our Q2Q pipeline. Note that we currently have to rely on our synthetic datasets, which allows us to evaluate the feasibility of our approach. We want to highlight that the absolute numbers must therefore be treated with caution. All networks have been trained with datasets of size 1 million over 10 epochs. For testing, we generated additional datasets of size 100k.

Dataset Analysis. For the NERQ dataset, we generated questions and queries about 277 hotels and 3,486 restaurants in Singapore. Table 1 shows the sizes of vocabularies. The larger vocabulary size of the NERQ dataset is due to 50% of queries being randomly selected queries from the AOL dataset. In this respect, the difference in vocabulary size is smaller than anticipated. Note that 2,192 words in the NERQ dataset appear both within and outside of POI names, as many names contain or are completely comprised of dictionary words, such as *The Soup Spoon* or *Fragrance Hotel*.

QC and NERQ Processor. We evaluated both the QC and NERQ processor using the NERQ dataset. For QC, if all terms in a query were labeled with `O`, we annotated the query as `generic` and otherwise—that is, the query contained a POI name—as `cqa`. Table 2 shows the result in terms of recall, precision, and F1 score. In particular, the results for QC are very good, which can be expected because we currently consider only two classes. However, also the NERQ processor performs very well despite the introduction of noise into queries (see Section 6.3). Whether these very good results will hold up for real-world datasets is one of the fundamental questions we have to address in the long run (see a discussion in Section 8).

Q2Q Translator. We first evaluated the translation of queries into questions using the bilingual evaluation understudy (BLEU) algorithm [49], yielding a corpus-level score of 54.1. Although scores greater than 50 generally reflect a good translation, we observed that on a query-question pair level, many pairs yield a low BLEU score, although the predicted question is semantically equivalent to the reference question. For example, the two questions "Is there a long waiting line at <POI> at the moment?" and "Does <POI> have a long waiting line now?" express the same information need but only yield a BLEU score of 24.8. This underlines the inherent limitations of BLEU [6]. We therefore conducted a crowdsourcing experiment where we let workers rate how the predicted question q^{pred} and the reference question q^{ref} match a given query. For this, we selected the 300 queries with the lowest BLEU scores between q^{pred} and q^{ref}. To these 600 query-question pairs, we added 300 random pairs as negative samples. For quality assurance, we last added an equally distributed mix of 100 manually answered query-question pairs. Using Figure 8 (https://www.figure-eight.com/), we asked workers if the match between a query and a question is

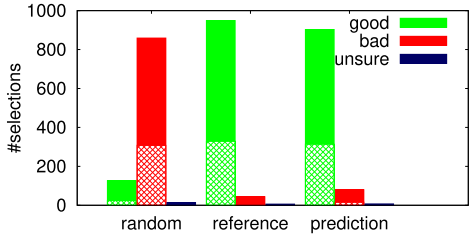


Fig. 5. Crowdflower results for Q2Q translation with respect to the three types of query-question pairs.

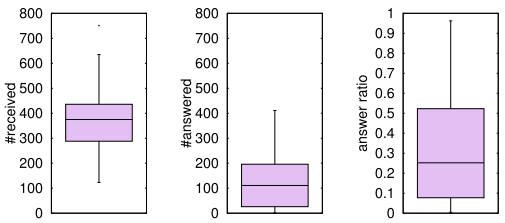


Fig. 6. Distribution of the number of received and answered private questions over all users.

Table 3. Classification with Examples of the 84 (100%) Questions Translated by Our Q2Q Pipeline That Have Been Labeled as *Bad* by at Least One Crowdflower Worker Regarding the Corresponding Queries

Category	**Examples**		
	#	Query	Question
Wrong grammar 33 (39.3%)	1	lobby strong smell POI	*Does the lobby in POI have a POI?*
	2	now POI average rate lowest	*Is POI average?*
	3	POI front desk quick	*Is the front desk at POI at?*
Odd phrasing 5 (6.0%)	4	front desk service able POI ability	*Is the front desk at POI able?*
	5	desk staff helpful administrators POI	*Is there a helpful desk staff at POI?*
	6	place POI now long waiting line	*Is the waiting line at the place inside POI long?*
Perceived error 31 (36.9%)	7	foyer awful laughable POI	*Is POI awful?*
	8	fifo long queue now POI	*Does POI have a long queue right now?*
	9	pretty music nice jazz now POI	*Is there nice music at POI at the moment?*
Arguably OK 15 (17.8%)	10	now POI size queue	*Does POI have a long queue right now?*
	11	size club lounge POI	*Is there a big club lounge at POI?*
	12	POI pool area nice areas located	*Is there a nice pool area at POI?*

Good, *Bad*, or *Unsure*; each question has been evaluated by three different workers. Figure 5 shows the results. Each bar shows the raw number of selected options (higher value) and the number of decisions based on majority voting (lower value). Most importantly, the distribution of results are very similar for the reference and predicted questions. In quantitative terms, the similarity between the results for the reference and predicted questions is 95.1%. This shows that most predicted questions reflect the intent of a search query well, even if the BLEU score is rather low.

For a more detailed error analysis, we looked at all 84 query-question pairs translated by our Q2Q pipeline that have been labeled as *Bad* by at least one Crowdflower worker. After a careful inspection, we assigned these 84 (100%) pairs to four categories (Table 3 presents three examples for each category)—note that language and writing style are often subjective, so the assignments are to some extent subjective as well: 33 (39.3%) of pairs feature questions that are grammatically wrong. Five (6.0%) pairs feature questions that, although grammatically correct, contain rather uncommon phrasing that might have resulted in the label *Bad*. Thirty-one (36.9%) pairs are most likely labeled as *Bad* because of an perceived mismatch between query and question. For example, in Example 9, the query contains "jazz," which workers are likely to have perceived as important but which is missing in the question. This is caused by the data augmentation during query generation, which tries to add related words to basic queries (see Section 6.3). Last, 15 (17.8%) pairs feature questions

that are arguably good translations of their queries (see Examples 11 and 12). Recall that each of the 84 query-questions pairs has been labeled *Bad* by at least one worker and that the same pair might have been labeled *Good* by other workers.

7.2 Mobile App User Experiment

To gain insights into users' opinions and experiences with the CloseUp mobile app, we conducted a field study with each of our 36 participants using the app for least 2 weeks. As required, the study had prior been approved by the institutional review board (IRB) of the National University of Singapore. All participants signed a consent form that informed them about the details of the study, including the collected personal data (Facebook ID, first name, email address, current geolocation). Each participant was remunerated with $20. The amount intentionally did not depend on the number or ratio of questions answered. Note that we did not evaluate the correctness of answers in this study. This would have required a much more closed/observed setting.

Simulating an Active Community. We deployed two bots to generate content as the surrogate for a (large) community—participants were aware of that, and the bots were aptly named. A reply bot answered any public question submitted by the participants within 1 minute by randomly selecting one of the predefined answers. A question bot sent different types of questions, all with a TTL of 60 minutes. From 7 am until 11 pm, the bot submitted a public question every 6 minutes on average. If a participant (i.e., his or her phone) was considered still for at least 2 minutes, the question bot sent questions such as "Is <POI> very crowed right now?" with <*POI*> being replaced with the name of the POI the participant was closest to (see Section 4.2). To avoid "spamming," the bot never sent two questions about the same POI twice in a row. Considering that the number of questions highly depends on users' locations and movements, the question bot also sent POI-independent questions every 60 minutes on average, also only from 7 am until 11 pm. Questions such as "Is it currently very windy at your location?" do not require users to be close to any POI like a restaurant or a hotel and could basically always be answered by the participants.

Data Analysis. In our analysis, we focused on the participants' behavior when receiving private questions. Recall that the mobile app only notifies users in case of new private questions. Participants could also answer and submit own public questions to explore the full feature set. Figure 6 shows the distribution of the number of received and answered private questions, as well as the distribution of the ratio of answered questions, over all participants. In general, the number of received questions does not vary much due to the (partially) fixed timings of the question bot. Participants who received more questions tend to travel/commute more, resulting in more POI-dependent questions. The number of answered questions is not surprisingly much lower than the number of received questions and varies more among the participants. The distribution of the answer ratio normalizes the absolute values of received and answered questions, and therefore yields the fairest comparison between participants. The results show that participants' behaviors differ very noticeably. However, with a mean greater than 30% and median greater than 25%, participants were rather active with respect to answering private questions. Regarding the timeliness of answers, Figure 7 shows the distribution of durations between receiving and answering questions for the most active users ($\geq$50 submitted answers). Note the upper bound of 1 hour, as this was the TTL for private questions. The results show that most questions got answered within 15 minutes, many even within 5 minutes.

Questionnaire Results. We first asked the participants how they use their mobile phones when on the move to identify how likely users see new private questions in a timely manner. As Table 4 shows, most users keep their phone readily available, often directly in their hand or at least in an easy-to-reach place. Note that the "Silent" setting still uses a flashing LED to notify users. We then asked the participants about their usage and experience with the CloseUp mobile app (Table 5).

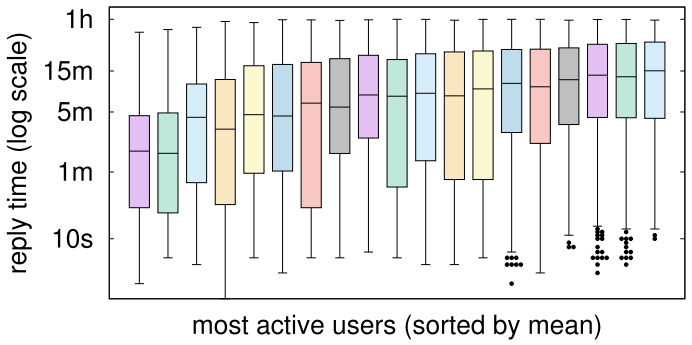


Fig. 7. Distribution of the time between receiving and replying to private questions for the most active users.

Table 4. Questionnaire Results for Participants' Basic Usage of Their Mobile Phones While on the Move

Where do you keep your phone when on the move? Answers: *Almost Never, Rarely, Sometimes, Often, Almost Always*					
Trouser or hip pocket	6%	3%	19%	19%	53%
Bag, backpack	13%	31%	28%	19%	9%
Armband (e.g., for exercising)	75%	13%	6%	6%	0%
Neck strap holder	84%	10%	3%	3%	0%
In the hand	0%	13%	28%	40%	19%
What are your phone's basic notification settings Answers: *Almost Never, Rarely, Sometimes, Often, Almost Always*					
Ringtone (with or without vibration)	27%	10%	27%	13%	23%
Vibration only	16%	9%	28%	19%	28%
Silent (no ringtone, no vibration)	9%	16%	28%	41%	6%

In general, the participants had no problems getting used to and using the app. Although some participants found the app annoying, that was caused by the settings of our questions bot. Second, most participants reported that they regularly answered private questions. This does not clash with an average answer rate of 25% to 30%, as app users are never aware of expired questions. In case participants did not willingly answer private questions, it was mainly because they could not answer and less because they did not want to. If participants could not answer, apart from being too busy, the main reason was that questions were not or no longer were relevant with respect to their current location. A commonly reported case was when participants were in a large shopping mall, where the accuracy of GPS, particularly in case of multilevel buildings, led to questions about POIs beyond participants' vicinity. If users did not want to answer questions, it was again mainly because they were occupied or felt that they already answered enough questions. Last, most participants expressed their wish for incentives to make answering more worthwhile. Apart from rewards in form of points, privileges, or even payments, the participants would also answer more frequently if the questions came from their circle of friends.

7.3 Web Search Interface User Experiment

After the field study for the mobile app, we introduced the CloseUp Web search interface to the same group of participants. As such, they were already familiar with the concept of live questions. We invited the participants to try out the interface and submit their own queries and questions.

Table 5. Questionnaire Results for Participants' Usage and Experience with the CloseUp Mobile App

How much do you agree with the following statements? Answers: *Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree*					
The app was intuitive & easy to use	0%	3%	9%	72%	16%
The app was (mostly) stable	0%	9%	19%	44%	28%
The app was annoying	3%	44%	19%	25%	9%
I find asking live question useful	6%	19%	28%	31%	16%
I would continue using such an app	3%	19%	38%	34%	6%
How often did you use the features of the CloseUp app? Answers: *Almost Never, Rarely, Sometimes, Often, Almost Always*					
Answering private questions	3%	9%	13%	41%	34%
Browsing public questions on the map	9%	25%	38%	16%	12%
Answering public questions	12%	16%	28%	25%	19%
Asking my own public questions	25%	25%	31%	16%	3%
Checking answers for my questions	19%	22%	28%	19%	12%
When you did not answer a private question, why? Answers: *Almost Never, Rarely, Sometimes, Often, Almost Always*					
I **could not** answer	6%	9%	25%	19%	41%
I **did not want** to answer	56%	16%	25%	3%	0%
*When you **could not** answer a private questions, why?* Answers: *Almost Never, Rarely, Sometimes, Often, Almost Always*					
The question was irrelevant	3%	25%	19%	19%	34%
I was too busy in that moment	12%	19%	25%	38%	6%
I was already to far from the POI	12%	19%	45%	12%	12%
I had technical problems with the app	66%	12%	16%	6%	0%
*When you **did not want** to answer a private questions, why?* Answers: *Almost Never, Rarely, Sometimes, Often, Almost Always*					
The questions was inappropriate	38%	28%	12%	16%	6%
I was occupied with other things	22%	9%	25%	28%	16%
I did not bother enough to answer	38%	22%	34%	6%	0%
I already answered many questions	28%	31%	13%	25%	3%
*When you **did not bother** to answer questions, what would motivate you?* Answers: *Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree*					
Karma points, badges, trophies, etc.	3%	19%	16%	40%	22%
Awarded privileges or features	3%	9%	13%	41%	34%
Real-word payments	3%	0%	9%	32%	56%
Virtual payments	6%	12%	19%	25%	38%
Questions from friends	6%	9%	16%	47%	22%

Our reply bot answered each question within 30 to 60 seconds, but participants could also answer their own questions using the mobile app. After that hands-on experience, we again asked them about their opinion about the interface (Table 6 presents the results). Most importantly, most participants see the benefit of live online search and would like if SEs such as Google would provide such features. Considering that questions with the Web search interface are (currently) sent anonymously, answers can only be displayed on the site itself. In this case, participants are not very willing to wait very long. In contrast, if answers were sent to the app or a dedicated portal, they would tolerate longer waiting times up to the TTL of the question. In general, the participants rated the quality of question suggestions together with the identified POIs as positive. We gave them no restrictions with respect to submitted queries. And although this was the first time where it was used by uninitiated users, it performed satisfactory in the majority of cases.

Table 6. Questionnaire Results for Participants' Opinion About and Usage of the CloseUp Web Search Interface

How much do you agree with the following statements? Answers: *Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree*					
Searching with questions is useful	13%	9%	19%	50%	9%
The Web interface is intuitive	6%	12%	20%	56%	6%
I wish Google offered this feature	6%	16%	6%	50%	22%
How long would you be willing to wait for answers? (TTL=60min) Answers: *<1min, 1-3min, 3-10min, 10-30min, 30-60min*					
Answers are shown in the interface	16%	34%	31%	19%	0%
Answers are sent to app (or email)	9%	19%	50%	19%	3%
Answers are updated on a portal	16%	22%	34%	22%	6%
How do you rate your experience with the search interface? Answers: *Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree*					
Questions matched my queries	9%	22%	32%	34%	3%
Expected questions were missing	3%	22%	40%	22%	13%
Places matched my query	6%	13%	22%	50%	9%
Expected places were missing	6%	22%	40%	16%	16%

8 DISCUSSION AND FUTURE DIRECTIONS

We implemented, integrated, and evaluated all core components of CloseUp. However, we do recognize its current limitations, and the necessity and potential for improvements. In the following, we outline a roadmap to advance community-driven live online search:

(1) Community building. The functionality of CloseUp is currently limited to the most basic features, such as handling questions and answers. Additional and potentially customizable or personalized features may improve the usability of the frontend applications, as well as the community aspect. Users might also be able to leave comments such as the latest special offers in a shop or restaurant. Meaningful features to enable socializing between users are friends lists or the support of a forum or even chat-like communication between users [69]. With its focus on time- and location-specific information, CloseUp creates a novel form of *community of practice* [68] where the shared knowledge is not permanent but typically becomes obsolete over time. How this characteristic effects users' behavior and thus the evolution of the community is an open research question.

(2) Incentivizing users. It is not obvious why users should reply to questions. Although this phenomenon has been observed in many traditional online communities, the number of active users in online forums or on Q&A platforms is typically still large enough for communities to be successful. In contrast, the number of people who can potentially answer a live question is typically rather low. It is therefore not clear if traditional incentive mechanisms like (karma) points, badges, additional features, or privileges (e.g., see [25, 32]) will be sufficient to establish an active community. A conceivable alternative is monetary rewards, such as giving users some cents for providing an answer. Although this might work in "closed" settings [31], it remains to be seen if such an approach is practicable and tractable on a large scale, or might even yield negative side effects [21].

(3) Veracity of answers. In general, a user asking a question has no reliable way to assess whether an answer is truthful or not. For example, a restaurant owner has an incentive to lie to attract potential customers. Traditional online communities, where a large user base can rate answers, are often self-correcting. This assumption does not hold for platforms like CloseUp. One potential solution is to deploy truth-telling incentives [58] like the *Peer Prediction* method [44] and *Bayesian Truth Serum* [51]. Another alternative is test questions [56], such as questions sent to users for

which the correct answers is known a priori. Such ground truth may stem from physical sensors. For example, consider a CCTV camera observing a public place capable of reporting the current level of crowdedness even without streaming the raw video feed [67]. Comparing users' answers with the camera's estimate can be used to update users' publicly visible reputation or trust score.

(4) Beyond Yes/No questions. One of our design goals was to make answering questions as quick as possible by selecting a predefined answer. This in turn requires providing a set of valid answers when submitting a question. Focusing on Yes/No questions made providing possible answers trivial. It also simplified the generation of questions since basic Yes/No questions feature, in general, a simple sentence structure. In our ongoing work, we investigate how we can automatically derive a meaningful set of possible answers for (more) arbitrary questions, such as WH questions (What, Where, When, Who, Why, etc.) and How questions. Note that this task is different from finding the true answer for a given question (e.g., see [24, 75]).

(5) Real-world datasets. Due to the lack of real-world datasets required for the training of the Q2Q pipeline, we put much effort into the generation of synthetic datasets. This approach allows evaluation of the feasibility and practicability of CloseUp. However, synthetic datasets limit the expressiveness and generalizability of the results. Additionally, to keep the complexity when generating well-form questions from exploding, we had to limit ourselves to Yes/No questions for now. We aim to utilize the mobile app, as well as existing crowdsourcing platforms, to collect real user questions, which then enable us to extrapolate from them to generate large-scale datasets.

9 CONCLUSIONS

Community-driven live online search is a novel paradigm toward social computing harnessing the wisdom of the crowds. It aims to close the blind spot of current SEs for finding live information best provided by mobile users. To demonstrate our vision of a Google-like and community-backed Web search experience, we designed, implemented, and evaluated CloseUp. Although being first and foremost a research prototype, we argue that the current version of CloseUp addresses all technical core challenges to bridge the gap between traditional online search and traditional CQA platforms. This particularly includes the automatic translation of keyword-based search queries into well-formed questions. The results of our user studies show that users find CloseUp both intuitive and useful. Apart from improving current methods and algorithms, the next steps will require addressing the nontechnical challenges of incentivizing users and ensuring the truthfulness of answers to build a large and stable community. To this end, we outlined our current research directions and potential solutions.

REFERENCES

[1] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of KDD'12*. ACM, New York, NY.

[2] Hazleen Aris and Marina Md. Din. 2016. Crowdsourcing evolution: Towards a taxonomy of crowdsourcing initiatives. In *Proceedings of the PerCom Workshops*. IEEE, Los Alamitos, CA.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.

[4] Cory Barr, Rosie Jones, and Moira Regelson. 2008. The linguistic structure of English Web-search queries. In *Proceedings of EMNLP'08*.

[5] Petter Bae Brandtzæg and Jan Heim. 2008. User loyalty and online communities: Why members of online communities are not faithful. In *Proceedings of INTETAIN'08*.

[6] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of BLEU in machine translation research. In *Proceedings of EACL'06*.

[7] Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys* 44, 1 (2012), Article 1.

[8] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of ICASSP'16*.
[9] Shuo Chang and Aditya Pal. 2013. Routing questions for collaborative answering in community question answering. In *Proceedings of ASONAM'13*. ACM, New York, NY.
[10] Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP'14*.
[11] Xiang Cheng, Shuguang Zhu, Sen Su, and Gang Chen. 2017. A multi-objective optimization approach for question routing in community question answering services. *IEEE Transactions on Knowledge and Data Engineering* 29, 9 (2017), 1779–1792.
[12] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP'14*.
[13] Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of NAACL-HTL'16*.
[14] Brooke Cowan, Sven Zethelius, Brittany Luk, Teodora Baras, Prachi Ukarde, and Daodao Zhang. 2015. Named entity recognition in travel-related search queries. In *Proceedings of AAAI'15*.
[15] Tom De Smedt and Walter Daelemans. 2012. Pattern for Python. *Journal of Machine Learning Research* 13 (2012), 2063–2067.
[16] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. 2011. Gamification. Using game-design elements in non-gaming contexts. In *Proceedings of CHI'11*. ACM, New York, NY.
[17] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. 2011. Crowdsourcing systems on the World-Wide Web. *Communications of the ACM* 54, 4 (2011), 86–96.
[18] William F. Eddy. 1982. *Convex Hull Peeling*. Physica-Verlag HD.
[19] Andreas Eiselt and Alejandro Figueroa. 2013. A two-step named entity recognizer for open-domain search queries. In *Proceedings of IJCNLP'13*.
[20] Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. 2013. BigBench: Towards an industry standard benchmark for big data analytics. In *Proceedings of SIGMOD'13*. ACM, New York, NY.
[21] Uri Gneezy and Aldo Rustichini. 2000. Pay enough or don't pay at all. *Quarterly Journal of Economics* 115, 3 (2000), 791–810.
[22] Çaglar Gülçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the ACL*.
[23] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of SIGIR'09*. ACM, New York, NY.
[24] Jiahui Guo, Bin Yue, Guandong Xu, Zhenglu Yang, and Jin-Mao Wei. 2017. An enhanced convolutional neural network model for answer selection. In *Proceedings of WWW'17 Companion*.
[25] Ferry Hendrikx, Kris Bubendorfer, and Ryan Chard. 2015. Reputation systems. *Journal of Parallel and Distributed Computing* 75, C (2015), 184–197.
[26] T. Hoßfeld, M. Hirth, P. Korshunov, P. Hanhart, B. Gardlo, C. Keimel, and C Timmerer. 2014. Survey of Web-based crowdsourcing frameworks for subjective quality assessment. In *Proceedings of MMSP'14*. IEEE, Los Alamitos, CA.
[27] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Synthetic data and artificial neural networks for natural scene text recognition. arXiv:1406.2227.
[28] Jiahua Jin, Yijun Li, Xiaojia Zhong, and Li Zhai. 2015. Why users contribute knowledge to online communities: An empirical study of an online social Q&A community. *Information and Management* 52, 7 (2015), 840–849.
[29] Xiao-Ling Jin, Zhongyun Zhou, Matthew K. O. Lee, and Christy M. K. Cheung. 2013. Why users keep answering questions in online question answering communities: A theoretical and empirical investigation. *International Journal of Information Management* 33, 1 (2013), 93–104.
[30] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. arXiv:1607.01759.
[31] Thivya Kandappu, Nikita Jaiman, Randy Tandriansyah, Archan Misra, Shih-Fen Cheng, Cen Chen, Hoong Chuin Lau, Deepthi Chander, and Koustuv Dasgupta. 2016. TASKer: Behavioral insights via campus-based experimental mobile crowd-sourcing. In *Proceedings of UbiComp'16*. ACM, New York, NY.
[32] Aikaterini Katmada, Anna Satsiou, and Ioannis Kompatsiaris. 2016. *Incentive Mechanisms for Crowdsourcing Platforms*. Springer.
[33] Joachim Kimmerle, Ulrike Cress, and Friedrich W. Hesse. 2007. An interactional perspective on group awareness: Alleviating the information-exchange dilemma. *International Journal of Human-Computer Studies* 65, 11 (2007), 899–910.

[34] Cliff Lampe, Rick Wash, Alcides Velasquez, and Elif Ozkaya. 2010. Motivations to participate in online communities. In *Proceedings of CHI'10*. ACM, New York, NY.
[35] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of AAACL'16*.
[36] Myriam Leggieri, Christian von der Weth, and John Breslin. 2015. Using sensors to bridge the gap between real places and their Web-based representations. In *Proceedings of ISSNIP'15*. IEEE, Los Alamitos, CA.
[37] Chenliang Li, Aixin Sun, Jianshu Weng, and Qi He. 2013. Exploiting hybrid contexts for tweet segmentation. In *Proceedings of SIGIR'13*. ACM, New York, NY.
[38] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of HLT'11*.
[39] Yefeng Liu, Todorka Alexandrova, and Tatsuo Nakajima. 2013. Using stranger as sensors: Temporal and geo-sensitive question answering via social media. In *Proceedings of W WW'13*. ACM, New York, NY.
[40] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP'15*.
[41] Monica Marrero, Julian Urbano, Sonia Sanchez-Cuadrado, Jorge Morato, and Juan Miguel Gomez-Berbis. 2013. Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards and Interfaces* 35, 5 (2013), 482–489.
[42] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13*.
[43] George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11 (1995), 39–41.
[44] Nolan Miller, Paul Resnick, and Richard Zeckhauser. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51, 9 (2005), 1359–1373.
[45] Mohamed Musthag and Deepak Ganesan. 2013. Labor dynamics in a mobile micro-task market. In *Proceedings of CHI'13*. ACM, New York, NY.
[46] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes* 30, 1 (2007), 1–20.
[47] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of CoNLL'16*.
[48] Jessie Ooi, Xiuqin Ma, Hongwu Qin, and Siau Chuin Liew. 2015. A survey of query expansion, query suggestion and query refinement techniques. In *Proceedings of ICSECS'15*. IEEE, Los Alamitos, CA.
[49] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL'02*.
[50] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Proceedings of InfoScale'06*. ACM, New York, NY.
[51] Dražen Prelec. 2004. A Bayesian truth serum for subjective data. *Science* 306, 5695 (2004), 462–466.
[52] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL'09*.
[53] Soumya Ray, Sung S. Kim, and James G. Morris. 2014. The central role of engagement in online communities. *Information Systems Research* 25, 3 (2014), 528–546.
[54] Ju Ren, Yaoxue Zhang, Kuan Zhang, and Xuemin Shen. 2015. Exploiting mobile crowdsourcing for pervasive cloud services: Challenges and solutions. *IEEE Communications Magazine* 53, 3 (2015), 1–9.
[55] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. 2012. Finding expert users in community question answering. In *Proceedings of WWW'12 Companion*. ACM, New York, NY.
[56] Dominic Seyler, Mohamed Yahya, Klaus Berberich, and Omar Alonso. 2016. Automated question generation for quality control in human computation tasks. In *Proceedings of WebSci'16*. ACM, New York, NY.
[57] Nigel Shadbolt, Max Van Kleek, and Reuben Binns. 2016. The rise of social machines: The development of a human/digital ecosystem. *IEEE Consumer Electronics Magazine* 5, 2 (2016), 106–111.
[58] Aaron D. Shaw, John J. Horton, and Daniel L. Chen. 2011. Designing incentives for inexpert human raters. In *Proceedings of CSCW'11*. ACM, New York, NY.
[59] Yikang Shen, Wenge Rong, Nan Jiang, Baolin Peng, Jie Tang, and Zhang Xiong. 2017. Word embedding based correlation model for question/answer matching. In *Proceedings of AAAI'17*.
[60] Ivan Srba and Maria Bielikova. 2016. A comprehensive survey and classification of approaches for community question answering. *ACM Trans. Web* 10, 3 (2016), Article 18.
[61] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS'14*.
[62] Luke Taylor and Geoff Nitschke. 2017. Improving deep learning using generic data augmentation. arXiv:1708.06020.
[63] Jaime Teevan, Kevyn Collins-Thompson, Ryen W. White, Susan T. Dumais, and Yubin Kim. 2013. Slow search: Information retrieval without time constraints. In *Proceedings of HCIR'13*. ACM, New York, NY.

[64] Trang Tran and Mari Ostendorf. 2016. Characterizing the language of online communities and its relation to community reception. In *Proceedings of EMNLP'16*.
[65] Rajan Vaish, Keith Wyngarden, Jingshu Chen, Brandon Cheung, and Michael S. Bernstein. 2014. Twitch crowdsourcing: Crowd contributions in short bursts of time. In *Proceedings of CHI'14*. ACM, New York, NY.
[66] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proceedings of ICML Deep Learning Workshop'15*.
[67] Yuhui Wang, Christian von der Weth, Thomas Winkler, and Mohan Kankanhalli. 2016. Tweeting camera: A new paradigm of event-based smart sensing device: Demo. In *Proceedings of ICDSC'16*. ACM, New York, NY.
[68] Etienne Wenger. 2011. Communities of practice: Learning, meaning, and identity. Cambridge University Press.
[69] Christian von der Weth, Ashraf M. Abdul, and Mohan Kankanhalli. 2017. Cyber-physical social networks. *ACM Transactions on Internet Technology* 17, 2 (2017), Article 17.
[70] Ryen W. White, Matthew Richardson, and Wen-Tau Yih. 2015. Questions vs. queries in informational search tasks. In *Proceedings of WWW'15 Companion*. ACM, New York, NY.
[71] Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computing* 1, 2 (1989), 270–280.
[72] Ian H. Witten, Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. (3rd ed.). Morgan Kaufmann.
[73] Haocheng Wu, Wei Wu, Ming Zhou, Enhong Chen, Lei Duan, and Heung-Yeung Shum. 2014. Improving search relevance for short queries in community question answering. In *Proceedings of WSDM'14*. ACM, New York, NY.
[74] Tingxin Yan, Matt Marzilli, Ryan Holmes, Deepak Ganesan, and Mark Corner. 2009. mCrowd: A platform for mobile crowdsourcing. In *Proceedings of SenSys'09*. ACM, New York, NY.
[75] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of NAACL'13*.
[76] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2011. A survey of crowdsourcing systems. In *Proceedings of PASSAT'11*. IEEE, Los Alamitos, CA.
[77] Yuxiang Zhao and Qinghua Zhu. 2014. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers* 16, 3 (2014), 417–434.

Received January 2018; revised October 2018; accepted November 2018