Understanding How People Use Natural Language to Ask for Recommendations

Jie Kang
University of Minnesota
kangx385@umn.edu

Kyle Condiff
University of Minnesota
cond0155@umn.edu

Shuo Chang[*]
Quora, Inc.
schang@quora.com

Joseph A. Konstan
University of Minnesota
konstan@umn.edu

Loren Terveen
University of Minnesota
terveen@umn.edu

F. Maxwell Harper
University of Minnesota
max@umn.edu

ABSTRACT

The technical barriers for conversing with recommender systems using natural language are vanishing. Already, there are commercial systems that facilitate interactions with an AI agent. For instance, it is possible to say "what should I watch" to an Apple TV remote to get recommendations. In this research, we investigate how users initially interact with a new natural language recommender to deepen our understanding of the range of inputs that these technologies can expect. We deploy a *natural language interface* to a recommender system, we observe users' first interactions and follow-up queries, and we measure the differences between speaking- and typing-based interfaces. We employ qualitative methods to derive a categorization of users' first queries (objective, subjective, and navigation) and follow-up queries (refine, reformulate, start over). We employ quantitative methods to determine the differences between speech and text, finding that speech inputs are typically longer and more conversational.

CCS CONCEPTS

• **Human-centered computing** → *Natural language interfaces*; • **Information systems** → *Recommender systems*;

KEYWORDS

recommender systems; natural language recommenders; critiquing; voice; speech; virtual assistants; qualitative methods; user study.

ACM Reference format:
Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. 2017. Understanding How People Use Natural Language to Ask for Recommendations. In *Proceedings of RecSys '17, Como, Italy, August 27-31, 2017,* 9 pages.
https://doi.org/10.1145/3109859.3109873

1 INTRODUCTION

"Hey Netflix, I want to watch an action movie ... spy-thriller ... maybe something like the Bourne Identity ... no, something less violent and more intellectual ... OK, show me the preview for Argo."

The above quote exemplifies an interaction pattern that we believe is coming to recommender systems in the near future. The user speaks to a device — in this case, a television screen — which incorporates each statement into a recommendation request. The request is matched against the user's preferences and globally-known properties of the database to find the most relevant results. As the results are presented on-screen, the user evaluates the results (potentially helping to clarify what he or she is looking for, which may be fuzzy) and issues a follow-up request to help the system find better results.

Several of the requirements to build a system like this are already in place. For example, set-top boxes from Amazon, Apple, and Comcast already listen for users to issue verbal commands or search for content. Some even allow users to ask for recommendations, though these features are currently incredibly shallow and limited. (Asking the Apple TV for "what should I watch tonight" results in a non-personalized list of content with the response "I hear these are worth checking out".) The piece that is missing from the current generation of natural language recommenders like these is the deep one: the ability to understand the nuanced intention of recommendation requests and to translate that intention into the right results.

This research serves to address a basic gap in our knowledge: though we can contrive examples like the one above to demonstrate the nuance of user requests, we do not know how people will actually speak or type to a fluent natural language recommender system. Therefore, we conducted and report on a study of users' *first interactions* with a new natural language recommendation-finding feature built into an established recommender system. By structuring our study around these first interactions, we seek to understand the goals that users might have, the types of queries they might issue, or how they might choose to express follow-up queries to refine results. To build natural language recommenders that respond to a full range of queries, we must begin to understand what the full range might be.

This paper is structured as follows. We first describe related work, with a focus on conversational recommenders, user goals in information retrieval, and prior comparisons of typing and speech. We then describe a prototype natural language recommender and an experiment that we used to collect recommendation-seeking

[*] Affiliated with the University of Minnesota during this project.

RecSys '17, August 27-31, 2017, Como, Italy
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4652-8/17/08…$15.00
https://doi.org/10.1145/3109859.3109873

requests. We follow this with three sections that constitute our primary contributions: (a) a qualitative analysis of recommendation-seeking goals in first queries, (b) a qualitative analysis of follow-up requests, and (c) a quantitative analysis of differences between text and speech modalities. We conclude with a discussion of the implications of these findings for designers of natural language recommenders.

As a further contribution, we make our experimental dataset publicly available [14]. In conducting this research, we collected recommendation-seeking queries, follow-up queries, and survey responses from 347 users of an established recommender system. This dataset is the first of its kind, and can be used to seed system-building or to facilitate subsequent studies on natural-language recommender systems.

2 RELATED WORK

One of the directions of this work — to explore the question of how voice and text compare in recommendation-seeking requests — builds on prior results in comparing text with speech. Most relevant is an analysis of the semantic and syntactic differences between spoken and typed-in queries using mobile search engine logs [9]. That work reveals that spoken queries are "closer to natural language", and are more often phrased as questions. Research on mobile search has contributed other relevant results, including a related finding that spoken queries are more "natural" and longer [6], a frequency analysis of mobile search categories [13], and a categorization of voice transcription errors and subsequent query reformulation strategies [12]. Early CSCW work investigated the differences between speech and text in annotating documents, finding that speech led to more effective communication of higher-level concepts, while text was superior at low-level comments [3]. Recent work has shown that technological barriers to voice input and transcription are disappearing [23].

Surprisingly little research has been done on the use of natural language (typed or spoken) in recommender systems. There has, however, been substantial work around the related topic of "conversational recommenders" [21, 26], where the user and the system engage with one another to iteratively refine a query. For instance, researchers have explored "critiquing" interfaces that offer users the chance to offer suggestions like "More like item A, but cheaper" [15]. However, little of this work involves natural language. One notable exception is the Adaptive Place Advisor [8], an early natural language conversational recommender, where the user and the recommender system engage in a natural text-based dialogue to narrow down a set of restaurant recommendations. This work was later extended to become a spoken dialogue recommender system based on a personalized user model [28].

In this work we explore how people use natural language to express a recommendation-seeking goal. Several notable recommender systems papers have proposed a set of goals or intentions that recommender systems should support. For instance, the human-recommender interaction (HRI) model [16] proposes terminology for describing a user-centric view of a recommender system, and is based on a hierarchy of user goals and tasks. This work explicitly recognizes that users are not always able to fully express their information needs; they label this uncertainty factor "concreteness". One of the most highly-cited papers on evaluating recommender systems [10] proposes a set of user tasks that categorize user goals in a recommender system; those goals (e.g., "find good items" and "just browsing") are at a higher level of abstraction from the goals studied here. In general, recommender systems research has focused more on supporting these high-level tasks, as opposed to designing for or understanding lower-level, goal-driven recommendation seeking tasks [25].

The field of information retrieval (IR) has contributed more research on understanding lower-level information seeking goals. Several highly-cited papers contribute categorizations of how users interact with search engines. Perhaps the most influential categorization [2] proposes three types of user goals in search: "navigational" (to reach a particular site), "informational" (to acquire information), and "transactional" (to locate services). Rose and Levinson extended this understanding by manually coding a set of 1,500 queries from AltaVista, adding a second level of categorization, and describing a frequency analysis of the different user goals [22]. Subsequent work used machine learning methods to infer these user goals across seven search datasets [11].

3 DATA COLLECTION

To investigate the use of a natural language recommendation-seeking interface, we built an experimental voice-based search interface into MovieLens[1], an established movie recommendation site, and asked site members to use it and provide feedback. This section describes the context of the study, the methods for collecting user responses, and the attributes of the final dataset that we use for subsequent analysis.

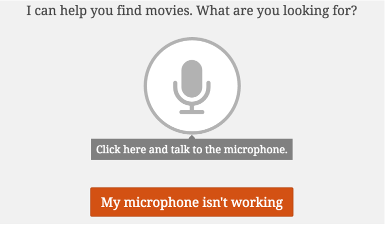
3.1 Experimental Site

MovieLens provides personalized movie recommendations for its members. The user experience is largely oriented around the process of finding movies to watch: members rate the movies they've seen to receive personalized recommendations based on those ratings. MovieLens does not offer any sort of natural language search features, though the site has a prominent "omnisearch" widget that allows users to search for a title, tag, or person.

We augmented MovieLens with an experimental interface that allows users to speak (or type) to the system. The system responds to the user request with a list of ten movies. To illustrate, the user might say "great car chases", and the system might return "Mad Max: Fury Road" and nine other movies that feature cars, chases, and/or greatness.

To be clear: building a state-of-the-art natural language query engine for movies is out of scope for this research project. Such a system would require substantial domain-specific investment at each level of the system architecture from voice recognition that understands actors' names to keyword extraction that is tuned to the particular needs of movie searches. Our goal here is a system that works "well enough" to allow us to answer our research questions — a method employed in prior HCI research, e.g. [30]. Though we considered deploying a "Wizard of Oz" recommender, we wish to promote ecological validity by giving subjects a system that "feels real" and can be used at the place and time of their choosing. To

[1] http://movielens.org



I can help you find movies. What are you looking for?

Click here and talk to the microphone.
My microphone isn't working

Figure 1: A screenshot of the voice input interface.

promote this validity, we recruited subjects with the promise of trying "an experimental feature" that we are considering bringing to MovieLens.

To build the prototype, we combine several off-the-shelf components with custom server-side search logic. On the client (web browser) side, we incorporate a voice input widget and voice-to-speech service from Wit.ai (https://wit.ai) to accept users' speech. The interface requires the user to click a button to start and end the process of collecting voice input. We immediately send the audio to wit.ai, which converts the audio to text. We allow users to view the results and to re-try or edit the results manually if the transcription results in errors, or if their microphone is not working.

We send the transcribed query to the MovieLens server to produce a list of the most relevant movies using custom search logic. Our process first uses AlchemyAPI (http://www.alchemyapi.com) to extract keywords from the query string, then searches for movies with titles, actors, directors, genres, or tags that match those extracted keywords. Movies with more matches get higher scores.

See the Data Collection Results section below for experimental evaluation of the quality of the recommender's results.

3.2 Subject Recruitment, Conditions, and Tasks

To evaluate user query behavior, we recruited MovieLens users by email. To bring in users with at least minimal familiarity with MovieLens, we emailed only users who had logged in during the previous six months and who had rated enough movies (15) to unlock personalized recommendations. Users who clicked on the link in the email were logged in to MovieLens and shown an experimental consent form; subjects were given the chance to immediately opt-out or stop at any time. This experiment was approved by our research institution's IRB.

We include just two experimental conditions: `speaking` or `typing`. Users in the speaking condition submit their queries by speaking at their computer or device, while users in the typing condition submit their queries by typing into an input box. Our assignment is not random, because some users do not have a working microphone attached to their computer. Therefore, our assignment is based on the strategy visualized in Figure 2. Subjects who cannot use a microphone are put in the typing condition. Subjects who can use the microphone are assigned randomly to the speaking condition (75% chance) or the typing condition (25% chance). There is a possibility that users with working microphones are different from users without working microphones (e.g., they might be more tech-savvy), and we examine this potential bias below.

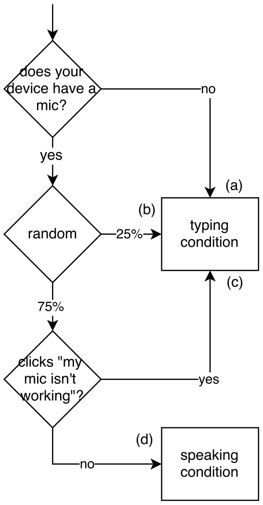


Figure 2: Method for assigning subjects to the `speaking` or `typing` condition. We first ask subjects if they have a working microphone. Users who say "no" are put into the typing condition (a), while users who say "yes" are randomly assigned to typing (b) or speaking. Users who are assigned to the speaking condition can click "my microphone isn't working" to fall back to the typing condition (c), or they can use the voice interface (d).

We ask consenting subjects to complete several tasks in sequence, summarized in Figure 3. All subjects begin the study by interacting with the typing- or speaking-based natural language input interface with the prompt "I can help you find movies. What are you looking for?" (see Figure 1). Once the subject submits a query, the system responds with ten search results and a short survey asking subjects to rate "how well do these results match what you were looking for" (on a 5 point "very poor" to "excellent" scale). Subjects who rate the results overall as "very poor" or "poor" are asked to explain how the results could be improved using free text input; subjects who rate the results as "fair" or better are asked to express a *follow-up query* (the interface prompts: "I can improve these results. Tell me more about what you want."). Finally, all subjects are surveyed about several factors, including how they hope the feature would work in MovieLens and their experience with other voice recognition interfaces. We state the specific wording of individual survey questions alongside results below.

3.3 Data Collection Results

We emailed 9,972 MovieLens members on May 12, 2016 and collected data through May 24; 544 consented to participate (5.5%). We collect at most one response per subject. For each subject, we consider their input to be "valid" if it is non-empty and not a testing or nonsense query (our coding methods are described below). For example, we exclude the user queries "testing one two three testing one two three" (testing), "xx" (nonsense), and "blah blah blah blah blah blah blah blah" (nonsense). Further, we discard any queries from the speaking condition where the user completely rewrote

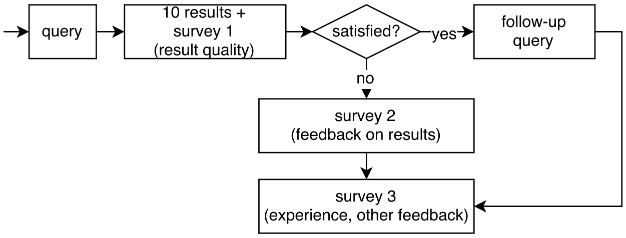


Figure 3: Flow chart describing subjects' experimental tasks.

Dataset	Total	Speaking
`input sample`	347	95 (27.4%)
`survey sample`	224	65 (29.0%)
`follow-up input sample`	151	43 (28.5%)

Table 1: Number of subjects (total, and in the speaking condition) in each dataset.

the transcribed speech using the text area, as it is not clear if these queries are representative of speech or text input. Resultantly, this excludes all users who received an empty transcription from wit.ai.

After these filtering steps, our dataset contains 347 valid queries, which we label the `input sample`. These queries were generally short in terms of character count (median 14, average 18.0) and word count (median 2, average 3.1). While 195/347 (56.4%) subjects reported having a working microphone, only 95/347 (27.4%) are in the speaking condition. This number is much lower than the expected value (~146) given the 75% random assignment rate; this is explained by two factors: (a) 40 users who manually switched into the typing condition by clicking the "my microphone isn't working" button after trying the microphone, and (b) a higher dropout rate in the speaking condition due to users who give up (and therefore do not click "send") after receiving one or more low-quality transcriptions.

We further divide subjects into two additional datasets to analyze survey responses and follow-up queries. 224/347 (64.6%) of the subjects provided complete survey responses, which we label the `survey sample`. 151/347 (43.5%) of the subjects provided a valid follow-up query, which we label the `follow-up input sample`. See Table 1 for an overview of the size of each dataset.

Overall, subjects tended to rate the quality of the MovieLens search results as "fair" — 156/224 (69.6%) of subjects reported that the results matched what they were looking for at fair or better, while the others reported "very poor" (12.9%) or "poor" (17.4%).

4 FIRST QUERIES

We code the queries from the `input sample` in several ways to facilitate frequency analysis and other forms of quantitative analysis on the dataset of transcribed user queries. Generally, we follow the inductive, open coding approach described in [18] for analyzing open-ended comments through constant comparison [7], which is based on grounded theory methods [1]. Specifically, four of the researchers involved in this project read through the dataset of user queries together, assigning new codes or refining old codes as we went. Our goal is to describe the queries in terms that will be useful to recommender system designers in our content domain.

Once we had developed a stable hierarchy of codes, two researchers separately coded 187 random responses to measure consistency and to help calibrate their coding practice (Cohen's kappa across 14 codes on 187 responses: avg=0.87; min=0.72; max=1.0). To determine the final coding used in this analysis, the two researchers then both coded *all* of the responses, then discussed and resolved disagreements.

4.1 Recommendation Goals

Inspired by classic work on categorizing web search [2, 22], we seek to understand users' underlying goals for interacting with the recommender. Similar to a search engine dedicated to searching across the web, a natural language-based interaction in a recommender system is a means to achieving a goal — in this case to find great movies to watch or to browse information about movies they are already interested in. Users express their queries in different ways in order to achieve this goal.

As with [22], our inductive coding method leads to a hierarchy of user goals, summarized in Table 2. We develop three top-level goals: objective, subjective, and navigation.

We define an *objective* goal as a request that can be answered without controversy. These goals seek to filter the movie space by specifying an attribute such as a genre, an actor, or a release date. These types of objective goals are often easy to answer using the sort of information that is typically available on movie websites. However, we find many examples of objective goals that cannot be easily answered using a typical database of movie information. We label these goals as seeking "deep features", indicating that users wish to filter movies by nuanced or specific criteria. Some examples of requests including deep features are "apocalyptic special effects" and "a movie about berlin wall".

We define a *subjective* goal as a request that involves judgment, uncertainty, and/or personalization. While objective goals tend to act as boolean filters (a movie either stars Brad Pitt or it does not), subjective goals are a more natural fit with a scoring or ordering algorithm. For instance, the query "interesting characters" might apply to many movies, some more strongly than others. Answering subjective queries — much like objective deep features — is difficult, because neither metadata databases nor recommender systems may track how much "clever plot" or "sad" a movie has.

We divide subjective goals into three common sub-types. Emotion requests tend to specify a particular feeling that a movie invokes in the viewer, e.g. "cheerful comedy". Quality requests are either explicit about wanting good/best movies (e.g., "Some *good* dystopic sci-fi would be nice."), or specify the aspects of the movie that make it good (e.g., "*classic* sci-fi movies'). Finally, movie-based requests seek related movies, e.g., "something like Pulp Fiction". We consider movie-based requests to be subjective rather than objective, because there is no objective and universally-held metric to determine the similarity between any two movies [5].

A *navigation* goal is the simplest of the three — the user wants to see one or more particular movies, so they state part or all of a title. Some examples in our dataset are "the social network" (which matches one movie) and "Star Wars" (which matches a series).

Recommendation Goal	Description	Examples
1. objective	My goal is to find movies based on their known, non-controversial attributes concerning...	
1.1 genre	...the type or category of movie	"superhero movies"
1.2 deep features	...uncommonly tracked features concerning plot, setting, or other nuanced characteristics	"movies with open endings or plot twists"
1.3 people	...the people who star in or participate in making the movie	"Brad Pitt"
1.4 release date	...when the movie was released	"can you find me a funny romantic movie made in the 2000s?"
1.5 region	...where in the world the movie is from	"british murder mystery"
1.6 language	...the primary language of the movie	"show me a list of german movies"
2. subjective	My goal is to find movies based on a quality judgment concerning...	
2.1 emotion	...the feeling of the movie	"sad movie"
2.2 quality	...the enjoyable parts of the movie	"interesting characters, clever plot"
2.3 movie-based	...the relationship to another movie	"what would you recommend to a fan of Big Lebowski?"
3. navigation	My goal is to find a particular movie by its title	"blade runner"

Table 2: Hierarchy of coded recommendation goals. A query can have more than one. E.g., "funny romantic movie made in the 2000s" codes as genre ("romantic"), quality ("funny"), and release date ("2000s").

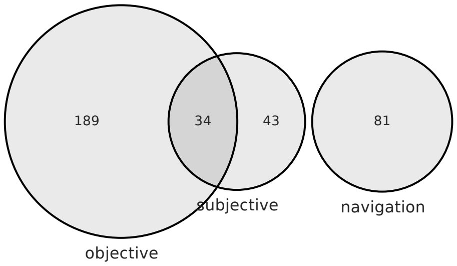


Figure 4: Venn diagram of top-level recommendation goals, which demonstrates how objective and subjective queries overlap, while navigation queries are independent.

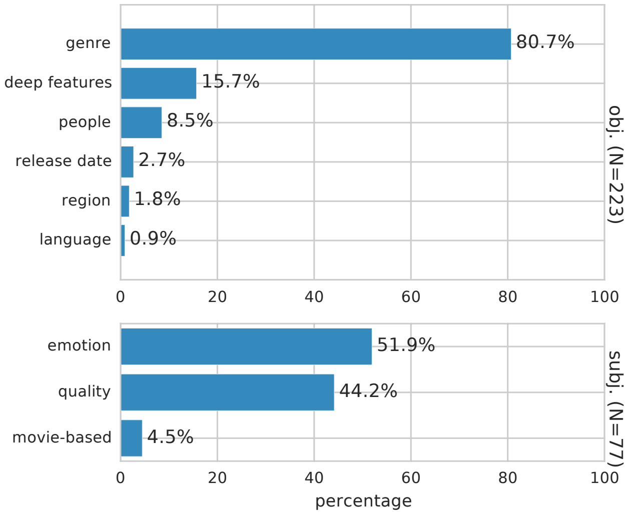


Figure 5: Percentage of objective (top) and subjective (bottom) requests containing the shown second-level goals. These goals are not mutually exclusive, therefore the bars in each chart sum to greater than 100%.

Unlike Rose and Levinson [22], our hierarchy of goals is not per-query exclusive; it is often the case that a single request contains several different goals. Users typically seek to intersect these multiple goals. For example, the query "drama movies with happy ending" combines two objective goals ("drama" genre, "happy ending" deep feature); the user wishes to find a movie with all of these qualities. We only observe two queries (2/343 = 0.6%) using the word "or", and only one of these appears to be requesting a union of multiple different goals ("movies with open endings or plot twists"). See Figure 4 for a visualization of the overlap between top-level goals, and Figure 5 for a visualization of the frequency of second-level goals in our data.

4.2 Other Coded Properties of Responses

We encoded several other properties of user responses that serve to improve our understanding of natural language queries. The coding methods, including the methods for measuring inter-rater reliability, are identical to those described above.

Conversational. Some queries are phrased as though the user is conversing with a human; we code these queries as conversational. Examples include "I'm looking for a hard sci-fi movie" and "find a movie like eternal sunshine of the spotless mind". These phrasings indicate a willingness to engage in a dialogue with a software agent. We found that 24.8% of the queries in our dataset are conversational.

Number of Modifiers. One measure of query complexity is the number of modifiers the query contains, where each modifier serves to filter or reorder the results. For example, "I'm looking for a movie that's not sad" has a single modifier ("not sad"), while "biographic dramas" has two ("biographic" [*sic*], "dramas"). Our earlier coding of recommendation goals is different from this metric because a request might have multiple components of the same type of goal (e.g., "spy thriller" and "adventure and drama" each contain two genre goals). In our dataset, 69.7% of the queries have a single modifier, 23.9% of the queries have two modifiers, 6.1% have three, and 0.3% have four (the maximum in our data).

Recommend. Some queries in our dataset explicitly seek recommended movies. Some examples are "a good movie" and "I'm looking for the best sci fi horror movie". Contrast these queries with the majority of queries where the desire for recommended movies is implicit ("looking for horror movies"), uncertain ("buddy cop movies" might seek the best movies in this genre, or a comprehensive view), or probably missing (navigation queries like "28 days later" seek a single movie, not a list). Only 4.4% of the queries in our dataset are explicit about seeking recommendations.

5 FOLLOW-UP QUERIES

Subjects who rate their search results as "fair" or better are asked to express a *follow-up query* with the prompt: "I can improve these results. Tell me more about what you want." This prompt is designed to elicit a second query, this time informed by a set of 10 search results. In this section, we qualitatively analyze queries from the subjects who complete this step (the `follow-up input sample`).

Unlike subjects' first queries, where their goals are typically explicit and recognizable, follow-up queries are commonly ambiguous with respect to their goals. For instance, a subject whose first query is "Science Fiction" and whose follow-up query is "Horror" could plausibly either be specifying an additional genre filter, or could be starting a new search.

Because of this ambiguity, in this analysis we adopt qualitative methods to identify *themes*, using word repetition and key-words-in-context methods [24]. Two coders went through the entire dataset to identify themes using these methods. The coders then discussed the emergent themes, integrated them, and extracted high-quality examples of each theme from the dataset.

In considering these findings, it is important to recognize that some subjects' initial query goals were met better than others. It is likely that the quality of the recommendations [20] affects subjects' follow-up behavior, but we do not investigate that link here.

5.1 Refine

Subjects commonly use the follow-up query to refine the initial query towards a more specific result (N=62, 41.1%). These subjects assume the system remembers their initial query, and specify *additional criteria* that they wish the recommender to consider.

Refine with further constraints. Many refinement queries suggest that the subject is still interested in the initial query, and wishes to further constrain the universe of the search space. Several examples follow (we denote the initial query with **1**, and the follow-up query with **2**):

1: a mystery drama with a suspenseful ending
2: something from the last few years

1: An action movie with a sense of humour
2: more sitcom less absurd

Refine with clarification. Other refinement queries reflect a disappointment with the initial results. These subjects attempt to help the digital assistant by providing more information:

1: Horror
2: More true horror instead of drama/ thriller

1: i'm looking for a great arts picture
2: this should really be an indie movie

5.2 Reformulate

Other subjects use the follow-up prompt to reformulate their initial query (N=34, 22.5%). These subjects appear to remain interested in their original query, but wish to *completely restate* the query to improve the recommendations. These subjects do not assume that the recommender remembers their last query, and typically reuse some portion of the original language.

Reformulate with further constraints. As with the refine queries, some subjects appear to reformulate to further narrow the set of results:

1: i'm looking for a romantic comedy
2: i'd like a romantic comedy that was created after the year 2000

1: i'm looking for time travel movie
2: I'm looking for a time travel movie that i haven't seen before

Reformulate with clarification. Other subjects reformulate queries in an attempt to encourage the system to better results:

1: a romantic comedy with a happy ending
2: romantic comedy with tensions between the couple but ends well

1: 28 days later
2: Movies like "28 days later"

5.3 Start Over

The third major theme we discovered in follow-up queries is that subjects want to start a new query (N=55, 36.4%), even though the experimental prompt says "*tell me more* about what you want" (emphasis not in the interface). These subjects may be experimenting with the system, or may realize that their first query is not at all what they are looking for:

1: mad max fury road
2: finding dory

1: red violin
2: new documentaries

6 SPEAKING VS. TYPING

The above analysis combines the recommendation-seeking queries from two modalities: speaking and typing. In this section we consider the differences in queries between these two modalities.

6.1 Subject Bias and Group Selection

We assign users in our experiment to either a `speaking` or a `typing` condition. However, our assignment procedure is non-random, as we wish to include users with no working microphone. Therefore, there are potential behavioral biases in our dataset between users in the different assignment pipelines. For instance, users who self-report no microphone might be more likely to be working on a desktop computer (vs. a portable device) or they might have less experience overall interacting with speech-activated interfaces.

See Table 3 for a summary of the number of users in each assignment category; the assignment process is shown above in Figure 2. There are three ways of reaching the typing condition, and only one way to reach the speaking condition.

We find evidence of several differences between users in the different assignment categories, which affects our subsequent analysis. We describe these findings in some detail as they reveal several interesting differences in self-reported experience and behavioral patterns. We test differences using a likelihood-ratio chi-squared test for categorical data, or a wilcoxon test for numerical data.

Subjects without a microphone self-report less frequent use of voice assistant technologies. Our survey contains the question "how often do you use a voice assistant (e.g., Google Now, Siri, etc.)?", with a six point response scale: "never" (0), "rarely" (1), "a few times a month" (2), "a few times a week" (3), "once a day" (4), "multiple times a day" (5). Subjects with a microphone (typing-random, typing-mic-not-working, and speaking-random) answered this question similarly (N=144, p=0.656). However, subjects without a microphone answered with lower scores (N=256, p<0.001). For instance, 50% of these users responded with "never" as compared with 21%-25% of subjects in the other three groups.

There are observable behavioral differences among the three assignment categories that feed into the typing condition. There are a few ways in which these three groups of subjects (a-c) behave similarly: they input approximately the same length queries (p=0.748), with a similar proportion of navigation (p=0.839) and objective (p=0.387) features. However, we find differences in the proportion of subjective features (typing-mic=10.0%, typing-mic-not-working=25.0%, typing-no-mic=28.9%; p=0.008), and conversational queries (typing-mic=6.7%, typing-mic-not-working=32.5%, typing-no-mic=19.1%; p=0.003). Subjects in group typing-mic-not-working also took longer (medians in seconds: typing-mic=17.5, typing-mic-not-working=42, typing-no-mic=20, p<0.001), though this may be explained by additional time spent determining the microphone failure and switching into the typing condition.

Due to these differences between assignment categories, we restrict this analysis to subjects in `typing-random` (N=60) and `speaking-random` (N=95). These groups are randomly assigned, and they report similar experience with voice assistant technologies (% subjects reporting 3 or higher: speaking-random=29.2%, typing-random=31.8%; p=0.386).

6.2 Results

Speaking to the recommender leads to longer queries than typing (medians in characters: speaking=19, typing=12.5; p<0.001). Related, we find that subjects in the speaking condition were much more likely to make conversational requests (proportion conversational queries: speaking=40.0%, typing=4.0%; p<0.001).

Speaking to the recommender takes more time (medians in seconds: speaking=39, typing=17.5; p<0.001). This effect may be due to time spent correcting transcription errors; possibly, the effect would disappear with a perfect voice recognition system.

Speaking leads to more queries with objective deep features (speaking=14.7%, typing=5.0%; p=0.047), and more subjective movie-based queries (speaking=5.3%, typing=0.0%; p=0.025). Other second-level type features have similar ratios between the two groups and do not show statistically significant differences in our dataset. Additionally, we did not find statistically significant differences between the two groups in terms of proportion of queries with objective (speaking=67.4%, typing=70%), subjective (speaking=17.9%, typing=10.0%), or navigation (speaking=22.1%, typing=23.3%) queries.

7 DISCUSSION

In this work, we use a prototype movie recommendation interface to learn more about how users might structure recommendation-seeking queries. Our work demonstrates both similarities and differences in how users approach recommendation and search. Our taxonomy of recommendation goals includes the concept of "navigational" queries that was developed to understand search behavior [2, 11, 27]. However, since recommendation is typically single-site, unlike web search, the other search goals from the information retrieval literature ("informational" and "transactional") are less applicable. Instead, we have chosen to model the characteristics by which users appear to filter and prioritize content — using "objective" and "subjective" criteria. Subjective queries, in particular, are interesting to consider, as they do a poor job of filtering, but provide an important signal to guide ranking. These differences highlight the shortcomings of applying current-generation search technology to the problem of natural language recommendations, and point to some of the key challenges that recommender systems researchers must overcome in developing next-generation systems.

Several prominent features of our recommendation-seeking taxonomy — namely, "objective deep features" and "subjective" features — are not easily handled by traditional recommendation algorithms. Deep features (e.g., "plot twists") are objective, but are out of the scope of entity metadata typically tracked. Subjective features (e.g., "great acting") are even more difficult, as they model opinion rather than objective truth. In our experiment, 15% of the queries contain "deep features", while 22% contain "subjective" features, underscoring their importance. One possible direction is the use of text-mining algorithms — such as the Tag Genome [29] or word2vec [17] — on unstructured text such as reviews. However, the prominence of these complex features points to a fascinating line of future work where systems combine the notion of "recommended items" with the user's contextual search for aspects such as "not violent" or "great acting".

In this research, we offered subjects the chance to follow-up their initial query, prompting "I can improve these results. Tell me

assignment category	N (input)	N (survey)	condition	has mic?	mic working?
(a) typing-no-mic	152	112	typing	no	N.A.
(b) typing-random	60	44	typing	yes	N.A.
(c) typing-mic-not-working	40	28	typing	yes	no
(d) speaking-random	95	72	speaking	yes	yes

Table 3: Four ways that users could be assigned to the typing or speaking condition. (a-d) correspond to the flowchart shown in Figure 2. Due to measured biases (explained in the text), we restrict our analysis of typing vs. speaking to groups b and d.

more about what you want." Given these instructions, we think it is surprising that many subjects issue "start-over" or "reformulate" queries. To explain this finding, we might look to information foraging theory [19] to indicate poor information scent [4] in the ten recommendations, or we might look to individuals' tendencies towards orienteering or teleporting search behaviors [27]. It is also possible that this behavior is a force of habit: frequent repeated use of search engines has trained us to understand that each query is a new query. We find that it is frequently difficult to disambiguate refine, reformulate, and start over queries, because the classification depends on the user's latent intent. Systems supporting natural language follow-up queries will have a difficult time inferring this intent, though it is critically important for determining the best recommendations. Interfaces or algorithms that naturally facilitate this disambiguation is an interesting area of future work.

While the future may bring nearly perfect voice recognition, the present offers tools that commonly make transcription errors. For ease of experimentation, to increase the pool of experimental subjects, and to avoid transcription errors such as these, researchers may wish to study natural language interfaces using typing as a surrogate for speaking. In this research, we find some key differences between these two modalities: speaking leads to longer, more conversational queries that are more likely to contain objective deep features ("plot twist") and subjective movie-based features ("movies like The Terminal"). Therefore, when text is used as a surrogate for speech, it will lead to somewhat different patterns of input, which may affect research results in some contexts.

7.1 Limitations and Future Work

Our work is based on the behavior of subjects who are seeking *movie* recommendations, and therefore it is unclear which of the findings presented here will generalize to other domains. We speculate that several of our high-level findings extend beyond the movie domain. For example, a variety of recommenders may find users expressing objective features to filter results along with subjective features to prioritize results. Also, there is nothing domain-specific to our finding that users more conversational in their queries when speaking than when typing. It is future work to confirm this speculation and to compare our current findings with natural language recommendation-seeking behaviors in other domains.

We cannot be certain that our small sample (N=347) is representative of the larger population of movie-recommendation seekers. Adding subjects from outside our experimental site – or simply expanding the pool of subjects from our site – might change the results of our frequency analysis or even reshape the outcome of our qualitative coding process. Our goal, given the scarcity of prior work on this topic, is to develop an initial, broad understanding of user behavior with a natural language recommender, which is served by a small sample. It is future work to expand this study to many users across sites to drill in on these results to further understand topics such as recommendation-seeking vocabularies or machine learning to infer recommendation-seeking goals.

Fundamentally, though we wish to explore an "open-ended" recommendation prompt, we found in internal testing that we could not simply use a Google-style interface (just an unlabeled box and a button) because users did not understand it. Therefore, we included a prompt designed to evoke a "virtual assistant" like Alexa or Cortana, stating "I can help you find movies. What are you looking for?" This prompt shapes the pattern of responses in an uncertain way. Future work might explore how different prompts — such as those employed by the Facebook Messenger-based movie recommender chatbot "And Chill" (andchill.io) — impact user requests.

8 CONCLUSION

In this paper, we describe a prototype natural language interface to a recommender system that prompts the user to open-ended recommendation requests. We study users' first interactions with the system to encourage users to express what they want, rather than the queries they know will work in a particular system.

We make several contributions to our understanding of recommender systems. To our knowledge, this is the first work to describe user recommendation requests in a natural language interface. To make sense of these requests, we contribute a taxonomy of user recommendation goals; the top-level goals are objective, subjective, and navigational. We also describe a dataset of follow-up requests, finding that while people use this second input to refine their initial request in a "critiquing" style, many others reformulate their query or start over. We study the differences between text and speech modalities, finding that speech leads users to longer, more conversational queries with more objective "deep features" and subjective "movie-based" features.

We collected a dataset of 347 users' first queries, follow-up queries, and survey responses, which we have released as an open dataset [14]; we hope this dataset will be a useful complement to this paper for systems-builders and researchers in developing the next generation of recommendation technology.

9 ACKNOWLEDGMENTS

This material is based on work supported by the National Science Foundation under grants IIS-0964695, IIS-1017697, IIS-1111201, IIS-1210863, and IIS-1218826, and by a grant from Google.

REFERENCES

[1] Andreas Bohm. 2004. TheoreticaI Coding: Text AnaIysis in Grounded Theory. In *A companion to qualitative research*. 270.

[2] Andrei Broder. 2002. A Taxonomy of Web Search. *SIGIR Forum* 36, 2 (Sept. 2002), 3–10. https://doi.org/10.1145/792550.792552

[3] Barbara L. Chalfonte, Robert S. Fish, and Robert E. Kraut. 1991. Expressive Richness: A Comparison of Speech and Text As Media for Revision. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. ACM, New York, NY, USA, 21–26. https://doi.org/10.1145/108844.108848

[4] Ed H. Chi, Peter Pirolli, Kim Chen, and James Pitkow. 2001. Using Information Scent to Model User Information Needs and Actions and the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 490–497. https://doi.org/10.1145/365024.365325

[5] Lucas Colucci, Prachi Doshi, Kun-Lin Lee, Jiajie Liang, Yin Lin, Ishan Vashishtha, Jia Zhang, and Alvin Jude. 2016. Evaluating Item-Item Similarity Algorithms for Movies. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2141–2147. https://doi.org/10.1145/2851581.2892362

[6] Fabio Crestani and Heather Du. 2006. Written Versus Spoken Queries: A Qualitative and Quantitative Comparative Analysis. *J. Am. Soc. Inf. Sci. Technol.* 57, 7 (May 2006), 881–890. https://doi.org/10.1002/asi.v57:7

[7] Jane Dye, Irene Schatz, Brian Rosenberg, and Susanne Coleman. 2000. Constant Comparison Method: A Kaleidoscope of Data. *The Qualitative Report* 4, 1 (Jan. 2000), 1–10.

[8] Mehmet H. Göker and Cynthia A. Thompson. 2000. Personalized Conversational Case-Based Recommendation. In *Advances in Case-Based Reasoning*, Enrico Blanzieri and Luigi Portinale (Eds.). Number 1898 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 99–111. DOI: 10.1007/3-540-44527-7_10.

[9] Ido Guy. 2016. Searching by Talking: Analysis of Voice Queries on Mobile Web Search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 35–44. https://doi.org/10.1145/2911451.2911525

[10] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53. https://doi.org/10.1145/963770.963772

[11] Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. 2008. Determining the informational, navigational, and transactional intent of Web queries. *Information Processing & Management* 44, 3 (May 2008), 1251–1266. https://doi.org/10.1016/j.ipm.2007.07.015

[12] Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How Do Users Respond to Voice Input Errors?: Lexical and Phonetic Query Reformulation in Voice Search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13)*. ACM, New York, NY, USA, 143–152. https://doi.org/10.1145/2484028.2484092

[13] Maryam Kamvar and Shumeet Baluja. 2006. A Large Scale Study of Wireless Search Behavior: Google Mobile Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 701–709. https://doi.org/10.1145/1124772.1124877

[14] Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. 2017. Understanding How People Use Natural Language to Ask for Recommendations: Query Dataset. (June 2017). http://conservancy.umn.edu/handle/11299/188687 type: dataset.

[15] Lorraine McGinty and James Reilly. 2011. On the Evolution of Critiquing Recommenders. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, 419–453. DOI: 10.1007/978-0-387-85820-3_13.

[16] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Making Recommendations Better: An Analytic Model for Human-recommender Interaction. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 1103–1108. https://doi.org/10.1145/1125451.1125660

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119.

[18] Hendrik Müller, Aaron Sedley, and Elizabeth Ferrall-Nunge. 2014. Survey Research in HCI. In *Ways of Knowing in HCI*, Judith S. Olson and Wendy A. Kellogg (Eds.). Springer New York, 229–266. DOI: 10.1007/978-1-4939-0378-8_10.

[19] Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological Review* 106, 4 (1999), 643–675. https://doi.org/10.1037/0033-295X.106.4.643

[20] Pearl Pu, Li Chen, and Rong Hu. 2011. A User-centric Evaluation Framework for Recommender Systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. ACM, New York, NY, USA, 157–164. https://doi.org/10.1145/2043932.2043962

[21] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, 1–35. DOI: 10.1007/978-0-387-85820-3_1.

[22] Daniel E. Rose and Danny Levinson. 2004. Understanding User Goals in Web Search. In *Proceedings of the 13th International Conference on World Wide Web (WWW '04)*. ACM, New York, NY, USA, 13–19. https://doi.org/10.1145/988672.988675

[23] Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James Landay. 2016. Speech Is 3x Faster than Typing for English and Mandarin Text Entry on Mobile Devices. *arXiv:1608.07323 [cs.HC]* (Aug. 2016).

[24] Gery W. Ryan and H. Russell Bernard. 2003. Techniques to Identify Themes. *Field Methods* 15, 1 (Feb. 2003), 85–109. https://doi.org/10.1177/1525822X02239569

[25] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. In *The Adaptive Web*, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Number 4321 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 291–324. DOI: 10.1007/978-3-540-72079-9_9.

[26] B. Smyth, L. McGinty, J. Reilly, and K. McCarthy. 2004. Compound Critiques for Conversational Recommender Systems. In *IEEE/WIC/ACM International Conference on Web Intelligence, 2004. WI 2004. Proceedings.* 145–151. https://doi.org/10.1109/WI.2004.10098

[27] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. 2004. The Perfect Search Engine is Not Enough: A Study of Orienteering Behavior in Directed Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 415–422. https://doi.org/10.1145/985692.985745

[28] Cynthia A. Thompson, Mehmet H. Göker, and Pat Langley. 2004. A Personalized System for Conversational Recommendations. *J. Artif. Int. Res.* 21, 1 (March 2004), 393–428.

[29] Jesse Vig, Shilad Sen, and John Riedl. 2012. The Tag Genome: Encoding Community Knowledge to Support Novel Interaction. *ACM Trans. Interact. Intell. Syst.* 2, 3 (Sept. 2012), 13:1–13:44. https://doi.org/10.1145/2362394.2362395

[30] Steve Whittaker, Julia Hirschberg, Brian Amento, Litza Stark, Michiel Bacchiani, Philip Isenhour, Larry Stead, Gary Zamchick, and Aaron Rosenberg. 2002. SCANMail: A Voicemail Interface That Makes Speech Browsable, Readable and Searchable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 275–282. https://doi.org/10.1145/503376.503426