March 2020

Obtaining Query Examples for Virtual Assistant Actions Using Search Logs

Asaf Zomet

Tomer Shmiel

Dvir Keysar

Amitai Cohen

Obtaining Query Examples for Virtual Assistant Actions Using Search Logs

ABSTRACT

A user's interaction with a virtual assistant typically involves the issuance of commands to perform certain actions, e.g., "play Mozart," "turn on the lights every night," etc. To execute actions accurately, a virtual assistant is trained to understand or detect action intent. This disclosure describes techniques to obtain a wide variety of query examples, with user permission, to train a virtual assistant to understand action intents. Information-seeking queries submitted to a search engine are transformed into action intents, e.g., commands, based on the grammar of the language of the search-engine query. The techniques are especially useful when launching new features, surfaces, or locales for a virtual assistant.

KEYWORDS

- Virtual assistant
- Spoken command
- Voice command
- Query translation
- Command translation
- Query intent
- Action intent
- Search log

BACKGROUND

A user's interaction with a virtual assistant typically involves the issuance of commands to perform certain actions, e.g., "play Mozart," "turn on the lights every night," etc. To execute actions accurately, a virtual assistant is trained to understand or detect action intent. If a user

command is a new use case for a virtual assistant or is in a new locale, few training examples (and their variants) are available: existing logs do not, by definition, cover new use cases or locales, and existing search-query logs are not typically formulated as action intents.

Training examples for virtual assistants can be obtained by crowdsourcing, e.g., different formulations of the same action intent can be obtained from native language speakers. However, this is slow, expensive, and does not always cover all the different ways by which users express intent. Alternatively, training examples can be obtained by translating queries from a well-supported language, e.g., English, to other languages. However, this depends on translation quality, does not cover the richness or jargon of the language, and does not work for new action intents. Still alternatively, different formulations of a given user intent can be obtained by rewriting existing examples with synonym tables. However, this can only expand the query in the dimension of the synonym and is restricted by the precision-versus-coverage tradeoff of the synonym, as synonyms can themselves be specific in intent.

DESCRIPTION

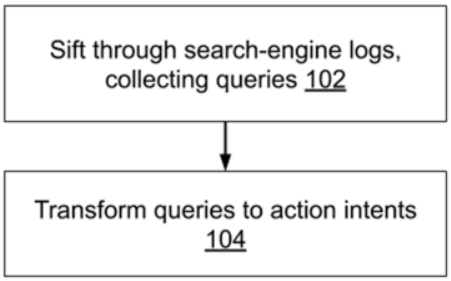

Fig. 1: Obtaining query examples for virtual assistant actions using search logs

Fig. 1 illustrates obtaining query examples for virtual assistant actions using search logs, per techniques of this disclosure. In this context, the phrases action intent, action-seeking query, and command are used interchangeably. With user permission, search engine logs are sifted

(102) for entries that are formulated as information-seeking queries. For example, queries such as “how to turn on the lights,” “how to play the guitar,” “how do I report a broken traffic light,” “what is a good time to visit NYC?” etc. that are formulated as questions are identified from search engine logs. The queries are transformed into action intents (104) based on the grammar of the language of the query. For example, in English, information-seeking queries can be transformed into action intents by simply stripping a leading suffix such as “how to,” as illustrated in the examples below.

Query	⇨	**Action intent**
“How to turn on the lights?”	⇨	“Turn on the lights.”
“How to play guitar like X?”	⇨	“Play the guitar like X.”
“How do I report a broken traffic light?”	⇨	“Report a broken traffic light.”

In this manner, training data for action intents is obtained by rewriting information-seeking search engine queries submitted by users to action-seeking queries or commands that are suitable for a virtual assistant. Variants of a query, e.g., “turn the lights on,” “turn on the lights,” “turn on the lights every night,” etc., that occur naturally in search engine logs are automatically included in the training corpus of action intents. In this manner, the repertoire of a virtual assistant can be rapidly expanded to include new locales and new use cases, e.g., car automation, ordering of food, etc. The query-formulation diversity found in search logs enriches action intents for virtual assistants.

```
germanToAction(howToPattern) {
const howToPrefixes = ['wie kann ich ', 'wie kann man '];

// Assume |howToPattern| begins with a prefix from |howToPrefixes|.
const prefix = prefixes.find(prefix => howToPattern.startsWith(prefix));
```

```
const tokens = howToPattern.substring(prefix.length).split(' ');

let verb = tokens.pop(); // verb should be last in 'wie kann...'

// Make verb imperative.
// Note: this is a basic conversion for German, which does not cover all
// imperative forms and exceptions. Can be further improved.
if (verb.endsWith('n')) {
verb = verb.slice(0, -1);
}

// Split to prefix+verb and leave separable prefix at the end.
const separablePrefixes = [
'ab', 'an', 'auf','aus','durch','ein',
'los','mit','vor','weg','zurück',
'zusammen','über','uber','um','unter','wider',
];

const verbPrefix =
    separablePrefixes.find((p) => verb.startsWith(p)) || '';

if (verbPrefix) {
tokens.push(verbPrefix);
}

tokens.unshift(verb.substring(verbPrefix.length));

return tokens.join(' ');
}
```

Fig. 2: Pseudo-code for query-to-action transformation for the German language

The query-to-action transformation, also known as rewrite rules, in a natural language depends on the grammar of the particular language. As seen above, in English, the query-to-action transformation can be as simple as removing a suffix or a prefix. For German, the query-to-action transformation is somewhat more involved and can be performed, e.g., as illustrated in the code snippet shown in Fig. 2.

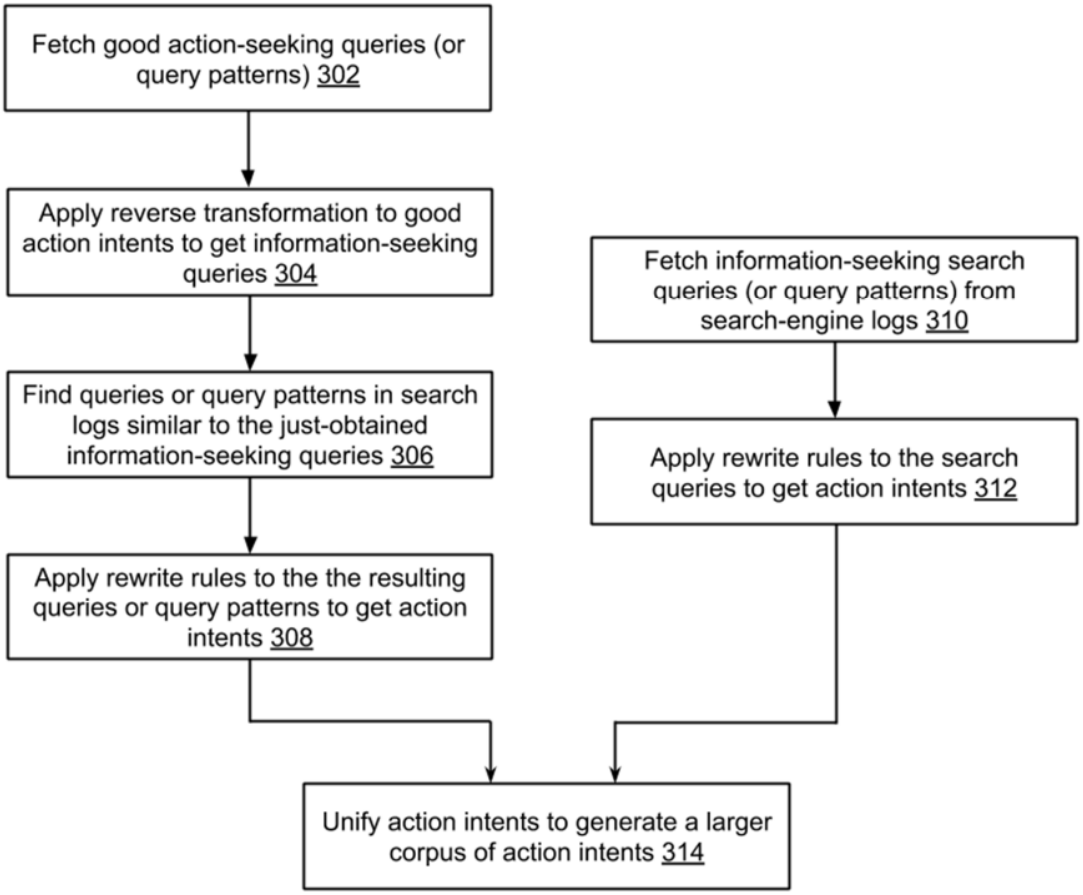

Fig. 3: Obtaining a larger corpus of action-seeking queries

Given the rewrite rules for a particular natural language, a larger training corpus of action-seeking queries can be obtained by generalizing good, action-seeking queries using the process illustrated in Fig. 3. A running example of generalization is illustrated with the action-seeking query “set a reminder.” With user permission, good, action-seeking queries or query patterns (“set a reminder”) are fetched (302). A reverse transformation is applied to the action-seeking query to obtain an information-seeking query (“how to set a reminder”). Unlike action-seeking queries, an information-seeking query such as “how to set a reminder” is likely to appear in search logs. The search logs are examined for queries or query patterns similar to the just-

obtained information-seeking queries (306), e.g., “how to put a reminder.” Rewrite rules are applied to the resulting queries or query patterns to get action intents (308), e.g., “put a reminder.”

In this manner, an action-seeking query “set a reminder” is generalized to other action-seeking queries, e.g., “put a reminder,” using the sequence of transformations “set to reminder” → “how to set a reminder” → “how to put a reminder” → “put a reminder.” Action-seeking queries are thereby generalized without their appearance in search logs.

On the other hand, as explained before, there may be information-seeking queries that regularly do appear in search logs, e.g., “how to turn on the lights.” As explained before, these are fetched (310) from search logs and transformed into action intents using rewrite rules (312).

A larger corpus of action-seeking queries is obtained by unifying the action intents (314) obtained by generalizing action-seeking queries and by applying rewrite rules to information-seeking queries.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user’s social network, social actions or activities, profession, a user’s preferences, or a user’s current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user’s identity may be treated so that no personally identifiable information can be determined for the user, or a user’s geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of

a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes techniques to obtain a wide variety of query examples, with user permission, to train a virtual assistant to understand action intents. Information-seeking queries submitted to a search engine are transformed into action intents, e.g., commands, based on the grammar of the language of the search-engine query. The techniques are especially useful when launching new features, surfaces, or locales for a virtual assistant.