November 2021

Techniques to Expand Grammar Pattern Matching

Amit Singhal

Michael Ng

Techniques to Expand Grammar Pattern Matching

ABSTRACT

Language understanding is an important aspect of supporting spoken queries on a device. Intensive ML models are unsuitable on resource-constrained devices and in such cases, pattern matching grammar can be utilized. However, pattern matching can have low recall and may be limited in understanding user queries. This disclosure describes the use of normalization, optional/stopwords, synonym clusters, and argument expansion to generate more expansive grammar with limited training data and without the use of machine learning. The use of such a grammar can improve recall while maintaining high precision. The techniques are suitable for any context where grammar patterns are used and where user queries are served by resource-constrained devices.

KEYWORDS

- Pattern matching
- Query pattern
- Query argument
- Query normalization
- Stopword
- Synonym cluster
- Voice assistant
- Virtual assistant
- Voice interface
- Voice query
- Car automation
- Vehicle infotainment
- Automatic Speech Recognition
- Natural Language Understanding

BACKGROUND

Language understanding is an important aspect of supporting spoken queries on a device. Machine learning (ML) techniques are commonly utilized for language understanding. However, given a limited training set, it is difficult to generate high quality and expansive grammar while managing device constraints. Intensive ML models are not available on resource-constrained devices. In such cases, pattern matching grammar can be utilized. However, pattern matching can have low recall and may be limited in understanding user queries.

On-device grammar models are typically generated from user-permitted traffic logs, obtained with user permission and anonymized. This allows for initial launch of new features in online (network-connected, server-supported) mode and offline (on-device) at a later time once enough traffic has been received. The use of query traffic is important to achieve high recall without the use of machine learning models, as traffic takes into account commonly spoken phrases. This technique is unsuitable when on-device grammar is to be launched simultaneously with online mode operation.

DESCRIPTION

It is important to expand the pattern matching capabilities on resource-constrained devices while maintaining high precision. This disclosure describes techniques to generate more expansive grammar with limited training data and without the use of machine learning. The techniques include normalization, use of optional/stopwords, synonym clusters, and argument expansion.

Normalization

For normalization, locale specific mappings of diacritics removal are enforced. Also, characters are changed to lowercase in both the grammar generation and on-device parsing layers. For example, in German, *ü* is mapped to *ue*. Enforcing normalization during both grammar generation and on-device parsing can eliminate issues that can arise from inconsistencies within the training set and speech recognition by forcing the use of normalized text. For example, if the training set has an example: “Bitte Lüftung an” and speech recognition techniques parse a user query as “Bitte lueftung an”, both are normalized to “bitte lueftung an”. Without normalization, the two queries would not match due to the difference in capitalization and diacritic use. Table 1 illustrates an example of improvements given the training query “Bitte Lüftung an”; as can be seen, enhanced matching that utilizes normalization succeeds where conventional matching fails

User Query	Original Match	Enhanced Match
Bitte Lüftung an	Yes	Yes
bitte lüftung an	No	Yes
Bitte lueftung an	No	Yes

Table 1

<u>Stopwords/ optional words</u>

Optional words or stopwords are words that are grammatically important for a sentence but do not add semantic value. For example, in the sentence: “Turn on the AC?”, “the” is a stopword as “turn on AC” conveys the same meaning. Generating a list of stopwords for an NLU engine allows marking stopwords in the grammar and user query as optional. The words contained within a training set are categorized as stopwords or important words. Such labeling of optional words in the grammar allows coverage over a large number of queries from a small set of training data.

Table 2 illustrates an example of the improvements enabled by a list of known stopwords (marked in red) given the training query: "set the temperature to 72 degrees"; as can be seen, enhanced matching that utilizes stopwords succeeds for queries that fail without the use of stopwords.

User Query	Original Match	Enhanced Match
Set the temperature to 72 degrees	Yes	Yes
Set temperature to 72	No	Yes
Can you please set the temperature to 72 degrees	No	Yes
Set my temperature to 72 degrees	No	Yes

Table 2

<u>Synonym clusters</u>

Words that are categorized as important words can be organized into synonym clusters. This allows words of similar meaning to be substituted in the grammar. For example, the terms "AC" and "Air Conditioning" have the same meaning. When added to the same synonym cluster, patterns containing any of the words can be matched to user queries containing either. By building synonym clusters and stopword lists specific to particular use cases, narrow patterns can be expanded to match dozens of different user queries. For example, when building grammar for car infotainment systems, synonym clusters and stopword lists specific to the domain can be utilized. Table 3 illustrates an example where the use of synonym clusters enables better matching. Given the training query: "increase the fan speed" and a synonym cluster that includes synonyms for "increase" (up, add, boost; shown in blue), and for "fan speed" (airflow, ventilation - shown in blue), query performance is enhanced. Note that the stopword "the" (shown in red) is also identified.

User Query	Original Match	Enhanced Match
Increase the fan speed	Yes	Yes
Up the airflow	No	Yes
Add fan speed	No	Yes
Boost the ventilation	No	Yes

Table 3

Argument expansion

Arguments within a grammar specify values that are used in query fulfillment. For example, queries for HVAC control within a car can have arguments that include absolute/relative values, location, temperature unit, etc. The query "set the temperature to 72 degrees Fahrenheit in the back" has 3 arguments: "72 degrees" as an *absolute value*, "Fahrenheit" as a *temperature unit*, and "back" as the *location*. In order to expand the training data patterns to accommodate a wider range of arguments, additional arguments are added wherever an argument is found.

For example, the training query "Set the temperature to 72 degrees" corresponds to the pattern "Set the temperature to ABSOLUTE_VALUE_ARG". In order to expand the grammar, additional optional arguments are appended to either side of the query like "Set the temperature to LOCATION_ARG? ABSOLUTE_VALUE_ARG TEMPERATURE_UNIT_ARG? LOCATION_ARG?. By building logical expansions within the context, the grammar is expanded in a meaningful way without causing overtriggering.

Table 4 illustrates an example where the training query: "Increase the temperature 2 degrees" with "2 degrees" as the *relative value* argument, is converted to the pattern "INCREASE_SYN the TEMPERATURE_SYN LOCATION_ARG?

RELATIVE_VALUE_ARG TEMEPRATURE_UNIT_ARG? LOCATION_ARG?" (synonym clusters shown in green and blue; stopwords in red; arguments in other colors). Such enhancement can improve the matching.

User Query	Original Match	Enhanced Match
Increase the temperature 4 degrees	Yes	Yes
Raise thermostat 2 degrees in the front	No	Yes
Increase the temp in the front 5 degrees Celsius	No	Yes

Table 4

Grammar enhancement via normalization, use of stopwords and synonym clusters, and argument expansion can improve recall substantially while maintaining high precision.

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 spoken queries, a user's context, user devices that receive spoken queries, 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

Language understanding is an important aspect of supporting spoken queries on a device. Intensive ML models are unsuitable on resource-constrained devices and in such cases, pattern matching grammar can be utilized. However, pattern matching can have low recall and may be limited in understanding user queries. This disclosure describes the use of normalization, optional/stopwords, synonym clusters, and argument expansion to generate more expansive grammar with limited training data and without the use of machine learning. The use of such a grammar can improve recall while maintaining high precision. The techniques are suitable for any context where grammar patterns are used and where user queries are served by resource-constrained devices.