

# Technical Disclosure Commons

---

## Defensive Publications Series

---

May 2020

## Automatic Issue Identification and Clustering

Petr Pilar

Mike Zuniga

Vladimir Yakovlev

Daniel Balle

Norbert Kalb

*See next page for additional authors*

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Pilar, Petr; Zuniga, Mike; Yakovlev, Vladimir; Balle, Daniel; Kalb, Norbert; Wenger, Stephan; and Baeuml, Martin, "Automatic Issue Identification and Clustering", Technical Disclosure Commons, (May 01, 2020) [https://www.tdcommons.org/dpubs\\_series/3201](https://www.tdcommons.org/dpubs_series/3201)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

---

**Inventor(s)**

Petr Pilar, Mike Zuniga, Vladimir Yakovlev, Daniel Balle, Norbert Kalb, Stephan Wenger, and Martin Baeuml

## **Automatic Issue Identification and Clustering**

### **ABSTRACT**

Search engines and virtual assistants receive a large number of queries, and also report a large number of issues such as user queries that didn't work, triggered the wrong features, or were not understood due to speech recognition errors, etc. This disclosure presents techniques that cluster similar issues, identify issues with the largest impact, and identify the root cause of an issue. The techniques scale easily, detect large patterns of similar issues, and prevent one-off fixes that need repeated application across similar issues. The techniques help improve the search engine or virtual assistant to provide responses that are more reliable, accurate, and satisfactory to the user.

### **KEYWORDS**

- Bug identification
- Issue clustering
- Issue identification
- Search engine
- Virtual assistant

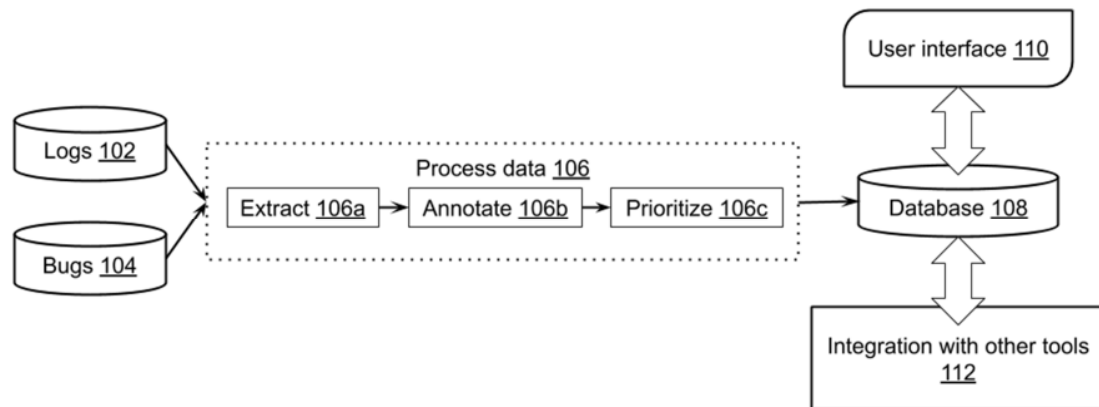
### **BACKGROUND**

Search engines and virtual assistants receive a large number of queries, and also report a large number of bugs or issues, e.g., user queries that simply didn't work (e.g., resulting in a response of the form "sorry I didn't understand that"), triggered wrong natural language understanding (NLU) features, were not understood due to errors in speech recognition, etc.

Lacking a programmatic clustering of similar issues, or the ability to pull in sufficient information that can serve to classify issues, search engine administrators today often manually

curate duplicated issues or focus on one-off solutions. The resulting inefficiency is compounded by the lack of clear prioritization, the lack of integration with other tools that actually fix the issue, and the lack of a user interface dedicated to the issue-clustering task.

### DESCRIPTION



**Fig. 1: Automatic issue identification and clustering**

Fig. 1 illustrates automatic issue identification and clustering, per techniques of this disclosure. Queries from search logs or queries (102) and bugs (104) are filtered based on whether the search engine (or virtual assistant) provided a good response or not. Query logs include only such queries and associated metadata (e.g., source device/application, query context, etc.) as permitted by the user. Users of virtual assistant or search engine software can turn off query logging entirely, or choose which of their queries are logged. Users are provided with information that indicates that query logs may be used for product debugging and improvement.

User queries with issues, e.g., queries that simply didn't work ("sorry I didn't understand that"), that triggered the wrong NLU features, that had errors in speech recognition, that were reported explicitly or implicitly by the user as being unsatisfactorily or incorrectly handled, etc., undergo processing (106), which includes the following:

- **Extraction of queries** (106a) based on the context in which the query and the failed response took place; locale; query source device or application (e.g., tablet, laptop, smart speaker, browser, virtual assistant); triggered feature; triggered domain; search vertical; previous queries; next queries; counts; failure rate; user-satisfaction scores; metadata; etc.
- **Annotation of queries** (106b), e.g., a semantic clustering of queries based on sentence embeddings or root-cause analysis; classifier-predicted features, domains, language, and search verticals; etc.
- **Prioritization** (106c), e.g., the impact of fixing the issue, based, e.g., on product considerations; query freshness; etc. Prioritization can also be done based on other factors such as severity, sensitivity, seasonality, etc.

Extracted, clustered, and prioritized queries are loaded into a database (108), from where these can be retrieved for display on a user interface (110) or used by other tools (112).

The extraction of mishandled queries can be done in various ways. For example, a query triplet in which the third query is a clarification of the first two queries is an indication of a mishandled query.

**Example:** Query 1 is “turn up the bedroom lights,” query 2 is “brighten the lights to 100%,” and query 3 is “brighten the bedroom lights to 100%.” Here, the third query is a clarification or combination of the first two queries, indicating that the first two queries were not handled in a satisfactory manner, and that all three queries are semantically related. If the third query is successfully acted upon, the clustering techniques described herein can improve future responses to the first two queries.

Semantic clustering of issues can be done using various criteria, e.g., by search vertical, root cause, intent, etc. For example, issue clustering can be persona-based; can be by grammar; by the semantics of the query relative to past queries; etc.

The user interface (UI), described in greater detail below, displays clusters of queries, arrays of classifier signals, etc. The UI enables an operator to further triage issues, e.g., by tagging, filtering issues, filing bug reports, etc.; to take action on an issue via integration with tools that improve grammars; to take action on an issue via integration with tools that improve responses in various search verticals; to annotate filters, clusters, groupings, etc.; to drill down to individual clusters and queries of interest or relevance; to improve responses to frequently mishandled queries; to persistently override clustering and classification; etc. Filtering of issues can be done by attributes, e.g., query source, locale, domain, predicted domain, root cause, predicted root cause, etc.

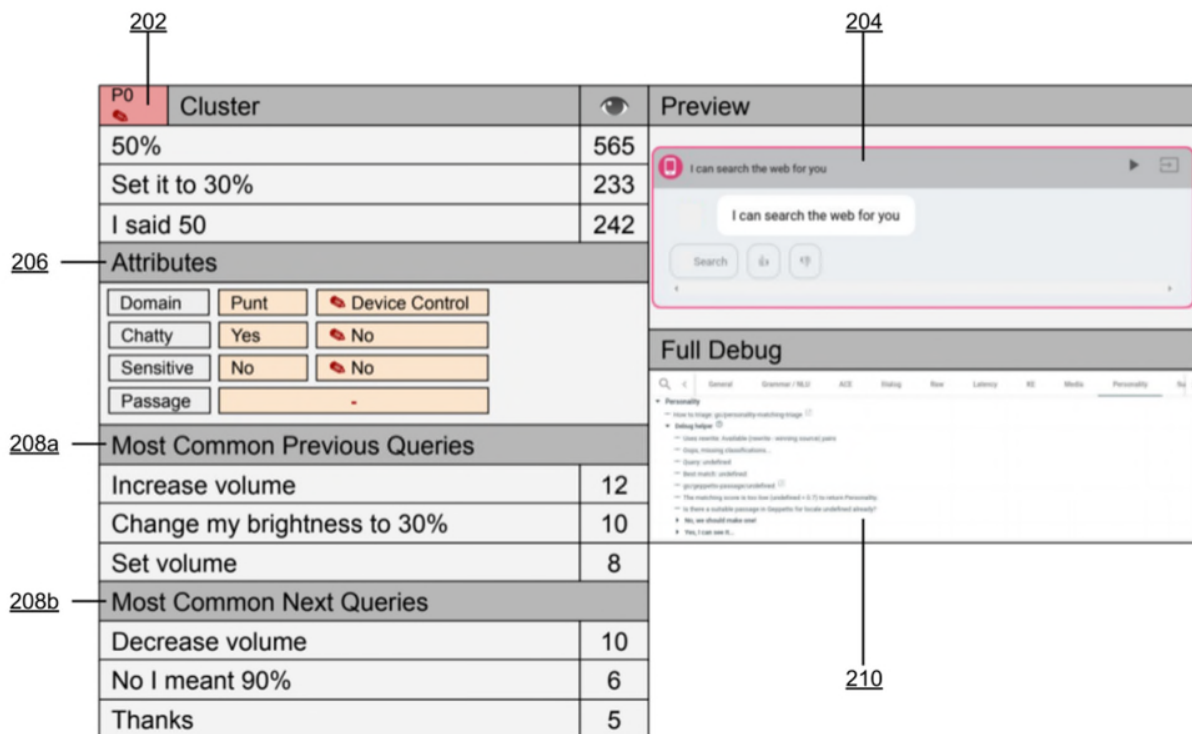


Fig. 2: An example user interface

Fig. 2 illustrates an example user interface. Clusters of queries that have failed or have been reported as unsatisfactorily handled are displayed (202). The user has the option to change (annotate) the name of such clusters. A preview of the actual assistant-user interaction is displayed (204), if available. Attributes of the issue-cluster are displayed (206), along with signals (208a-b) used for clustering. A full debug trace is offered (210) when available. Bug-fixing can be scheduled or queued across different engineering teams such that fixes do not conflict or trample with one another.

The disclosed techniques can be used by search engine or virtual assistant software providers, including team members such as engineers and operations teams to estimate the general headroom of a feature; to assess the impact of a given problem; to determine future product directions; etc. The advantages of the techniques include query clustering to identify and jointly address issues with semantically similar queries; the ease of integration with other tools and platforms; the presence of a user-friendly interface that can serve a range of professionals, e.g., engineers, product managers, operations personnel, marketing personnel, etc.

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 queries), 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

Search engines and virtual assistants receive a large number of queries, and also report a large number of issues such as user queries that didn't work, triggered the wrong features, or were not understood due to speech recognition errors, etc. This disclosure presents techniques that cluster similar issues, identify issues with the largest impact, and identify the root cause of an issue. The techniques scale easily, detect large patterns of similar issues, and prevent one-off fixes that need repeated application across similar issues. The techniques help improve the search engine or virtual assistant to provide responses that are more reliable, accurate, and satisfactory to the user.

## REFERENCES

[1] Gopal Jhavar, Madan, Vipindeep Vangala, Nishchay Sharma, Ankur Hayatnagarkar, Mansi Saxena, and Swati Valecha. "Conversational Help for Task Completion and Feature Discovery in Personal Assistants." [arXiv preprint arXiv:1907.07564](https://arxiv.org/abs/1907.07564) (2019).