Analyzing User Needs at NERSC with Natural Language Processing and Machine Learning Techniques

BERKELEY LAB

SUSTAINABLE RESEARCH PATHWAYS
SUSTAINABLE HORIZONS INSTITUTE

U.S. DEPARTMENT OF ENERGY
Office of Science

NERSC
National Energy Research Scientific Computing Center

Martha Asare[1], Dr. Lipi Gupta[2,3], Dr. Erik Palmer[2,3] , Dr. Brandon Cook[2]

[1]University of Texas Rio Grande Valley, [2]Lawrence Berkeley National Laboratory, [3]Mentor

ABSTRACT

The National Energy Research Science Computing Center (NERSC), a High Performance Computing facility, supports a large number of scientific projects whose users generate many support tickets due to the complexity of their tasks. Every year, NERSC receives over 6,500 user help tickets reporting issues and requesting services. In this study we explore the effectiveness of using Natural Language Processing and Machine Learning (ML) techniques on User ServiceNow ticket data to automatically identify frequent and repetitive problems that need attention. Starting with word frequency and moving to sentence embedding and clustering, we demonstrate the capability of each approach to generate actionable insights. Further development of these mechanisms will allow NERSC staff to make targeted enhancements and boost user research efficiency.

DATA SOURCE

- **ServiceNow user ticket data:** (Jan. 2023 - Jan 2024)
- **Total Tickets:** 6,944 with 24 fields
- **Sampled Data:** 3 fields of text data
  - **Help Ticket Title:** Summary of the ticket
  - **Additional Comments:** Added to help resolve the submitted tickets
  - **Comments and Work notes:** User and staff interaction on solving the submitted tickets
  - **Staff work notes:** Notes left on the ticket submitted for internal use

PRE-PROCESSING

- Combined all fields of text into 1 field
- Filtered out stop words using Python library ***"stopwords"*** and ***"WordNetLemmatizer "***.
- Retain timestamp of ticket created by user

```
'102551 globus remote endpoint connection failure 020015 – system (additional comment
resolved state. 104015 – [redacted] (additional comments) never mind case usual gl
file 102551 –[redacted] (additional comments) unable get perlmutter scratch via g
pdate related maintenance something else. message remote endpoint failure command fai
point nersc perlmutter [redacted] server [redacted]
l globus_xio gsi xio driver failed establish connection via underlying protocol.\\ngl
3\\nglobus_xio system error connect connection refused\\nglobus_xio system call faile
comment access globus often fraught intermittent resets. case unfamiliar error messag
le to/from perlmutter scratch needed. 020015 – system (additional comments) incident
ate. 104015 – [redacted] (additional comments) never mind case usual globus interm
–[redacted] (additional comments) unable get perlmutter scratch via globus right
d maintenance something else. message remote endpoint failure command failed command
perlmutter [redacted] server [redacted] message coul
gsi xio driver failed establish connection via underlying protocol.\\nglobus_xio unab
o system error connect connection refused\\nglobus_xio system call failed connection
```

Figure 1:A single sample of the processed ticket-text data from the 6,944 dataset.

RESEARCH OBJECTIVE

To analyze user behavior to identify common issues, in order to proactively assist users, and improve the user documentation.

KEY TERMS

- **User:** Any person using NERSC Perlmutter's computational resources
- **Application Terms:** A collections of names of software packages used at NERSC that run on the Perlmutter supercomputer.
- **Issue Terms:** A list of words that we have identified to describe common problems or request submitted by users.
- **Regex Matching:** Issue terms and Applications terms were matched concurrently using the Python library "re."
- **Sentence embedding** : A technique in NLP which transforms sentences to dense vectors of fixed or same length.

ANALYSIS TECHNIQUES

After preparing and cleaning the ticket text data we applied two main analysis techniques:

- Word Frequency Count
- Machine Learning Techniques

WORD FREQUENCY

We wrote code in Jupyter Notebooks running on NERSC's Perlmutter to process the text fields exported from the ticket database. From there, we constructed a custom Python kernel which allowed us to integrate several popular data analysis libraries, such as counter, matplotlib, sklearn.feature_extractor, etc . Using these tools we counted the occurrence of issue terms and application terms in each user ticket received.

```
"password": r"(?i)\bpassword\b",
"error": r"(?i)\berror\b",
"crash": r"(?i)\bcrash\b",
"configuration": r"(?i)\bconfiguration\b",
"backup": r"(?i)\bbackup\b",
"restore": r"(?i)\brestore\b",
"network outage": r"(?i)\bnetwork\boutage\b"
"log-in": r"(?i)\blog-in\b",
"activation": r"activation",
```

Figure 2: Partial list of regex expressions used in Jupyter Notebook for the study.

WORD FREQUENCY

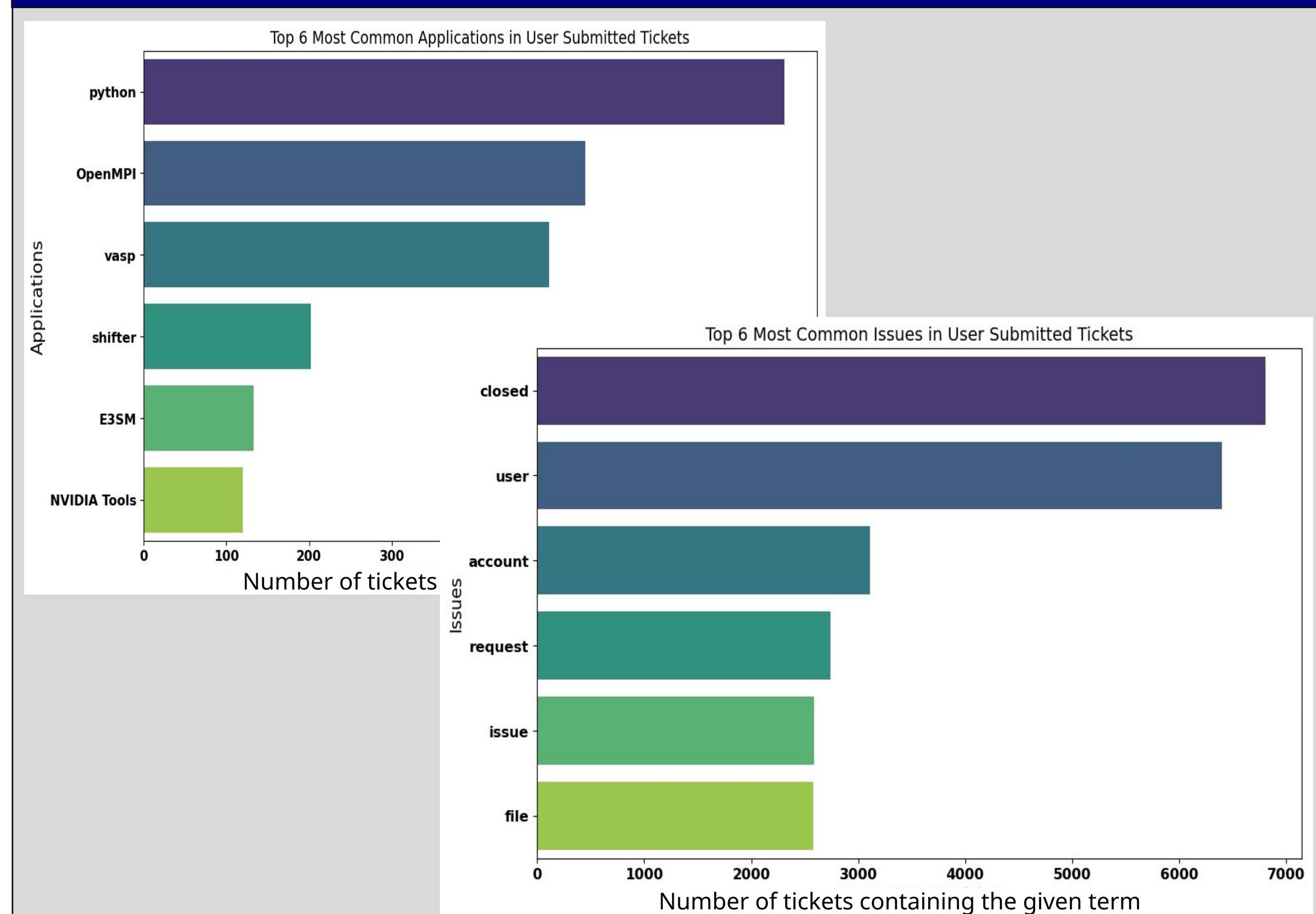

Figure 3 *Top-left*: Bar plot displaying the occurrence of top 6 issue terms in a user submitted ticket. Each term is only counted once per ticket. Figure 4 *Bottom-right*: Bar plot showing occurrences of top 6 issue terms.

By examining the correlation of words among application terms and issue terms we can generate a 2-D matrix that represents the amount tickets with both terms appearing together. The goal of this representation is to gain insight into what type of issues users have with each application.

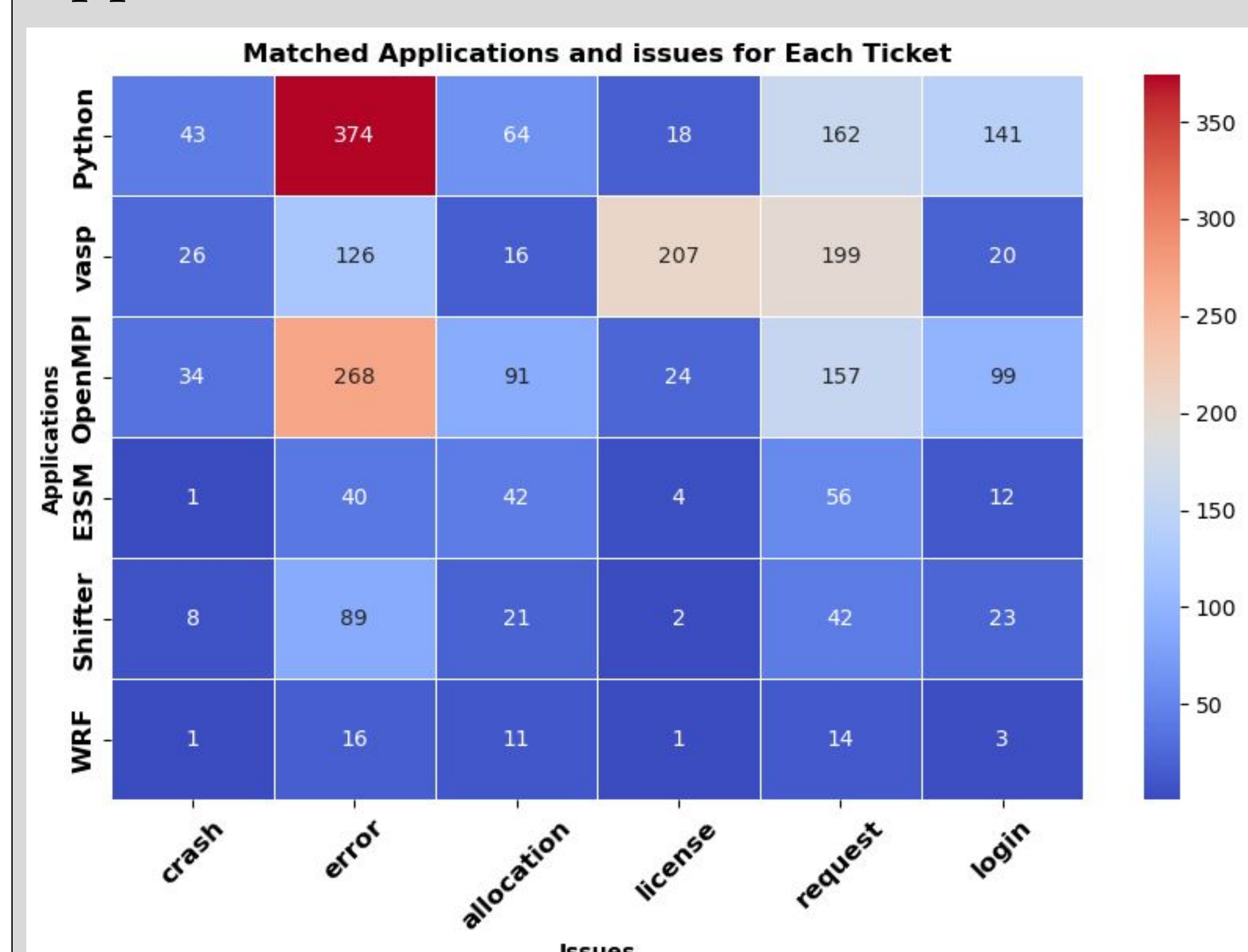

Figure 4 *Above*: Heatmap displaying the correspondence between application and issue terms for the most frequent applications and issues. Figure 5 *Below*: Heatmap of the entire data set (6,944) including all application and issue terms.

MACHINE LEARNING

We also explored if ML could effectively group tickets automatically. First we transformed the combined text fields as a "sentence" to a vector representation using the "all-MiniLM-L6-v2" language model. With this we were able to generate a similarity matrix that employs a cosine distance metric to determine similarity between two "sentences." Second, we used pyLDA[1] to generate topic clusters. From the visualized output we can draw conclusions about the ability of this approach to group tickets.

```
tensor([[1.0000, 0.3851, 0.4770,  ..., 0.4060, 0.2057, 0.5297],
        [0.3851, 1.0000, 0.7071,  ..., 0.3993, 0.3835, 0.4294],
        [0.4770, 0.7071, 1.0000,  ..., 0.3741, 0.3331, 0.4956],
        ...,
        [0.4060, 0.3993, 0.3741,  ..., 1.0000, 0.5850, 0.5037],
        [0.2057, 0.3835, 0.3331,  ..., 0.5850, 1.0000, 0.3764],
        [0.5297, 0.4294, 0.4956,  ..., 0.5037, 0.3764, 1.0000]])
```

Figure 5: 6,944 tickets were encoded as 384-dimensional vectors using a sentence transformer *with the pre-trained "all-MiniLM-L6-v2" language model.* The values in the matrix represent similarly between ticket-texts.

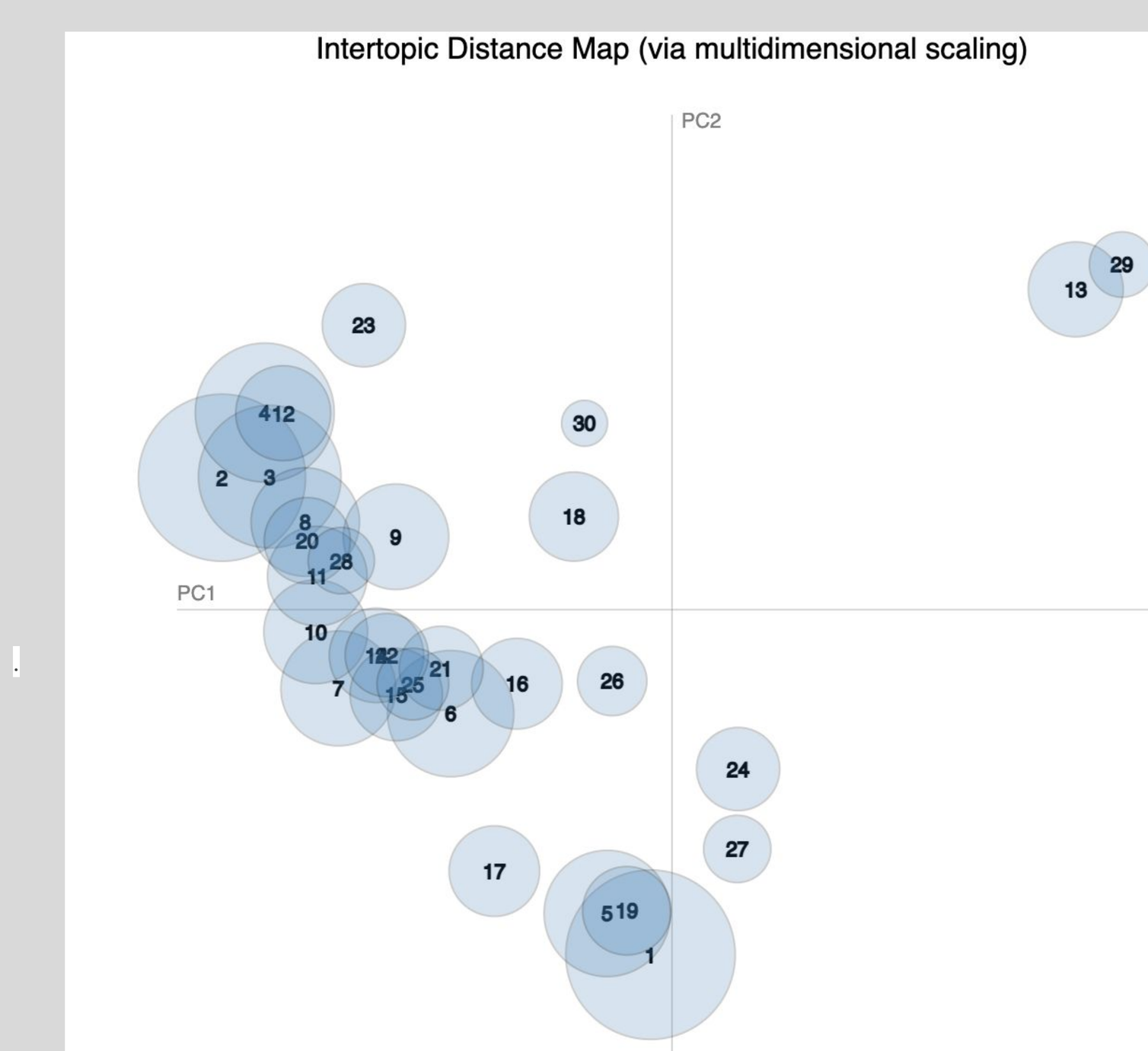

Figure 6: An Intertopic Distance Map generated with pyLDA and number of topics set to 30. The size of each circle represents the percentage of tickets assigned to that topic group. The position of the center of each circle represents its location along principal components determined by pyLDA and is useful for interpreting how similar or distinct two topics may be from one another. Circle centers with larger distances from others, indicate a more distinct topic.

[1]Sievert, C., & Shirley, K. (2014, June). LDAvis: A method for visualizing and interpreting topics. In Proceedings of the workshop on interactive language learning, visualization, and interfaces (pp. 63-70).

FUTURE WORKS

1. Improve text data cleaning to remove personal and standard system information.
2. Explore different cluster and topic modeling methods on high dimensional datasets.
3. Train ML algorithms for sentence transformation and clustering.
4. Use a Large Language Model to automatically label each topic cluster.

ACKNOWLEDGMENTS

I want to thank Sustainable Research Pathways (SRP), NERSC, and my mentors for the opportunity to work with LBNL. Though it was brief, the experience was incredibly valuable, providing me with real-life insights and skills that will significantly benefit my career.