T. Luke Banaszak
Tbanas2
CS410 - FA22

Final Project Proposal - IT Service Management System Improvements through Text Mining

Project Summary

My final project will demonstrate the value of applying multiple text mining methods within an IT service management system ("ITSM system"). The final product will be a proof of concept ITSM system with a frontend to collect new requests and backend that classifies new service requests based on the skill required to resolve them and sentiment analysis to identify angry responses for more urgent response.

Justification and Use Case

ITSM systems (aka ticket systems) are used to manage a technology support team's work by preserving work requests then allocating them across a team. They are the interface between users and the support team - users complete a webform requesting assistance which notifies the support team. These forms usually contain a subject, open text description, and additional criteria selected by the user (category, urgency, etc.).

Ideally, an ITSM system would help an IT support function self-sustain itself during a manager's absence by replicating the manager's delegation responsibility through a "if this then that" rules engine (i.e., "rules-based approach".) This is accomplished by pre-defining combinations of ticket forms and field selection values, and then specifying what group or individual that combination should route to for resolution. For example, if a ticket is submitted from the "hardware support" webform and marked as "low" urgency, it should be routed to the tier 1 hardware support team.

The rules-based approach has 2 primary shortcomings that hinder the desired self-management:

1. There are differing perceptions of issue categorization between the technology support group and the individual seeking assistance (i.e., the User).
    a. A user may be having a problem with their keyboard, but submit the help request under the "software" category because the issue presents an inability to use their web browser.
    b. The user might even just submit this request under whatever webform is most accessible and easy to use - they likely have little incentive to adhere to the manager's system of forms and rules and just "want it fixed".

2. The rule-based approach requires a high level of upkeep. Today's pace of change in technology has organizations constantly acquiring and decommissioning applications, and the rules engine needs to be manually updated every time this happens.

When submitting a help request, users enter a narrative description of their issue into the open-ended Description field. This is typically the most valuable information for the support team to understand the issue and who is the best team to resolve it. This information is open-text, natural language, though, so it cannot be used within the described rules-based approach.

Details on Technology, Data, and Time Estimates

My project will produce a simple web application that allows a user to enter a work ticket, and the backend will analyze the request and save it with the determined labels. This set of features represents the text analysis component of an ITSM system and produces the data that would be used by a routing capability to assign the requests based on skill.

- The system will be written in Python
- Sentiment analysis and classification are expected to be implemented using the NLTK or SKL Python libraries
- The sentiment analysis model will be trained using a generic corpus
- The classification model will be trained using an anonymized dataset from my job containing a ticket narrative and associated skill for resolution
- The frontend will use the Flask framework

The project's estimated time is:

- Cleaning and tagging my training and testing datasets: 4 hours
- Implementing sentiment analysis: 5 hours
- Implementing classification model: 5 hours
- Creating frontend of system and connecting to backend: 5 hours
- Database design: 2 hours
- Testing accuracy of labels: 3 hours
- Writing report and video: 5 hours