Valence Case: Labelling User-System Conversations

Milind Choudhary

[mxc210096@utdallas.edu](mailto:mxc210096@utdallas.edu)

**Task:**

To label and score the user-system conversations using LLM and Non-LLM based techniques

**Dataset Used:**

User Satisfaction Simulation - Conversational Dataset (<https://github.com/sunnweiwei/user-satisfaction-simulation>)

In particular, I used a subset of the **Schema Guided Dialogue (SGD)** dataset which contained annotated task-oriented conversations between humans and a virtual assistant across different domains.

**Evaluation Metrics:**

The metrics used were taken from the paper “[Simulating User Satisfaction for the Evaluation of Task-oriented Dialogue Systems](https://arxiv.org/pdf/2105.03748)”

* **Unweighted Average Recall:** UAR is the arithmetic average of all class-wise recalls.
* **Kappa Score:** It is used to check the inter-annotator agreement, but it can be modified to judge the predictions by evaluating how well the predicted labels agree with the true labels across all classes.
* **Spearman Rho Coefficient:** Spearman’s rank is used to find the relationship between two variables. It can be modified similarly to suit our task.

**Data Preparation:**

Analysis of the dataset:

1. The dataset was heavily imbalanced towards an average rating of 3
2. Voting which is essentially the mode was used as a tiebreaker between annotators which might not be correct always.
3. Statistical Analysis:

* Mean of gold labels: 3.09
* Median of gold labels: 3
* Mode of gold labels: 3
* Standard deviation of gold labels: 0.44
* Minimum score: 1
* Maximum score: 5

Preprocessing and Cleaning:

The code used in the repository was mostly used to preprocess and clean the dataset.

Steps involved:

1. Extracting turn-by-turn conversation from the input SGD.txt
2. Calculating the mode of every annotation by different annotators

**Non-LLM Based:**

Preprocessing for Non-LLM based:

* Conversion of the text into numerical values using TF-IDF which provides insight into how the words are spread across individual data points as well as across different documents.
* The length of the input text was also stored as a feature.

Models Used:

* Logistic Regression: LR was used as the most basic model to find the baseline of classification.
* Support Vector Classifier: This kernel-based approach is used a lot due to its ability to generalize using kernels that represent the input data in higher dimensions to make it linearly separable.
  + HyperParameters tuned:
    - Gamma
    - C
    - Kernel
* Random Forest: RF was used since it takes advantage of Decision Trees ability to segregate the dataset based on the features and also experiment with different weights and different architectures.
  + HyperParameters tuned:
    - Number of estimators (trees)
    - Depth of trees
    - Splits per node
    - Samples per leaf node
* XGBoost: XGBoost is used in the industry due to its feature to quickly adjust to the data points by weighing them based on misclassification.
  + HyperParameters tuned:
    - Number of estimators
    - Maximum Depth
    - Learning Rate
    - Gamma

Results:

1. Visual Representation of the experiments and fine tuning. (<https://wandb.ai/milindc02-university-of-texas-at-dallas/valence?nw=nwusermilindc02>)
2. Final Results
   1. UAR

A screenshot of a computer

Description automatically generated

* 1. Kappa

A screenshot of a graph

Description automatically generated

* 1. Spearmans Rho

A screenshot of a computer

Description automatically generated

**LLM Based:**

Preprocessing for LLM based:

* Converting the dataset into a json file to feed the OpenAI-based models.
* Functions to randomly select demonstrations to be provided to the GPT models
* Prompts made:

**P1 (0 shot first prompt):**

"""You are an expert linguistic assistant.

Your task is to label and give a score to conversations for user satisfaction. The score for the user satisfaction are based on a 5-level satisfaction scale.

The scale is as follows:

(1) Very dissatisfied (the system fails to understand and fulfill users request);

(2) Dissatisfied (the system understands the request but fails to satisfy it in any way);

(3) Normal (the system understands users request and either partially satisfies the request or provides information on how the request can be fulfilled);

(4) Satisfied (the system understands and satisfies the user request, but provides more information than what the user requested or takes extra turns before meeting the request); and

(5) Very satisfied (the system understands and satisfies the user request completely and efficiently).

You should predict only the score which is a number and print it as Score.

"""

**P2 (0 shot statistical input)**

"""You are an expert linguistic assistant.

Your task is to label and give a score to conversations for user satisfaction. The score for the user satisfaction are based on a 5-level satisfaction scale.

The scale is as follows:

(1) Very dissatisfied (the system fails to understand and fulfill users request);

(2) Dissatisfied (the system understands the request but fails to satisfy it in any way);

(3) Normal (the system understands users request and either partially satisfies the request or provides information on how the request can be fulfilled);

(4) Satisfied (the system understands and satisfies the user request, but provides more information than what the user requested or takes extra turns before meeting the request); and

(5) Very satisfied (the system understands and satisfies the user request completely and efficiently).

You should predict \*\*only the score\*\* and print it as `Score:` followed by the appropriate number.

Additionally, here is the statistical information from a dataset of labeled conversations to help you make informed judgments on how scores are distributed:

- \*\*Mean of gold labels\*\*: 3.09

- \*\*Median of gold labels\*\*: 3

- \*\*Mode of gold labels\*\*: 3

- \*\*Standard deviation of gold labels\*\*: 0.44

- \*\*Minimum score\*\*: 1

- \*\*Maximum score\*\*: 5

These statistics indicate that most of the conversations are labeled as \*\*3 (Normal)\*\*, with some variability. You should consider this when labeling satisfaction, but still make your score based on the specific conversation at hand.

"""

**P3 (shortened statistical)**

"""You are an expert linguistic assistant.

Your task is to label and give a score to conversations for user satisfaction. The score for the user satisfaction are based on a 5-level satisfaction scale.

The scale is as follows:

(1) Very dissatisfied (the system fails to understand and fulfill users request);

(2) Dissatisfied (the system understands the request but fails to satisfy it in any way);

(3) Normal (the system understands users request and either partially satisfies the request or provides information on how the request can be fulfilled);

(4) Satisfied (the system understands and satisfies the user request, but provides more information than what the user requested or takes extra turns before meeting the request); and

(5) Very satisfied (the system understands and satisfies the user request completely and efficiently).

You should predict \*\*only the score\*\* and print it as `Score:` followed by the appropriate number.

Most of the scores are 3 and the minimum value is 1 while the maximum value is 5. The scores have a standard deviation of 0.44 from the mean which is 3. Use this statistical information but do not be biased by it.

"""

Models Used:

* GPT 3.5 ($0.5/1M tokens)
* GPT 4o mini ($0.15/1M tokens)

Results:

1. Preliminary Testing:

A graph of different colored lines

Description automatically generated

1. Main Subset Dataset:

A graph with numbers and lines

Description automatically generated

**RAG Based:**

Preprocessing for LLM based:

* Converting the dataset into a json file to feed the OpenAI-based models.
* Using **Milvus** to create a vector store (Code attached)

Sentence Embeddings used:

1. Open AI embeddings
2. Sentence Transformers

Results:

* Results are in the graph attached with LLM-based
* RAG generalizes to the dataset quickly, but LLM has to be finetuned for better performance since it overfits.