QuickAssist

Final PPT

Team members:
Inbal Bolshinsky
Shani Kupiec
Almog Sasson
Nadav Margaliot

Repository

1.1 Project Description And Project Objectives

Project Title: QuickAssist: Intent-Aware Chatbot for Customer Support

Problem Statement:

- Many customer support bots fail to generate helpful or appropriate responses due to lack of intent understanding.
- Customer queries are often vague or ambiguous, requiring contextual inference.
- Objective: Improve chatbot response quality by conditioning on explicit intent labels.

Why It's Important:

- Effective customer support automation improves response time and user satisfaction.
- Poor responses can lead to user frustration or unresolved issues.

1.1 Project Description And Project Objectives

Why It's Challenging:

- Language is often inconsistent and domain-specific.
- Intents may be implicit or span multiple overlapping categories.

Project Objectives:

- Evaluate the impact of intent conditioning on response quality.
- Compare different modeling strategies:
  - Single-step vs. Two-step pipelines
  - Pretrained vs. Fine-tuned models
- Investigate generalization across datasets (Bitext and Customer-Service-for-LLM).

1.2 Formal Task Specification

Task Overview:

- **Input:**
  - Variant A: Customer query (free text)
  - Variant B: Customer query + intent label
- **Output:** Generated customer support response

Evaluation Metrics:

- **Automatic:** BERTScore (semantic similarity to reference), BLUE and ROUGE-L
- **Human-like:** LLM-based scoring of Helpfulness, Fluency, Appropriateness

1.2 Formal Task Specification

Subtasks Overview:

- **Data Preparation:** Load and merge two datasets with pre-annotated intents
- **Training:**
    - Intent Detection: Fine-tuned BERT or pretrained T5
    - Response Generation: Pretrained or fine-tuned T5
- **Evaluation:** Compare 6 configurations including:
    - single_step_pretrained, single_step_ft
    - two_step_baseline, two_step_pretrained, two_step_partial_ft, two_step_complete_ft

1.3 Prior Art

Source/Title	"Intent-Aware Dialogue Generation and Multi-Task Contrastive Learning for Multi-Turn Intent Classification," 2024	"RSVP: Customer Intent Detection via Agent Response Contrastive and Generative Pre-Training," 2023
Task Solved	Multi-turn intent classification and response generation	Intent detection from dialogue context
Approach/Model	Hidden Markov Models (HMMs) combined with Large Language Models (LLMs), Multi-task Contrastive Learning	Two-stage self-supervised model: retrieval and generative pre-training using agent responses
Data	E-commerce chat logs with domain-specific intent transitions (details not specified)	Task-oriented dialogue datasets, focusing on agent utterance-response pairs (details not specified)
Metrics	Intent classification accuracy; coherence and relevance in multi-turn conversations	Response retrieval accuracy; generation quality and relevance (exact metrics not specified clearly)
Results	Effective intent-driven response generation with contextually coherent conversations (precise numerical results not detailed explicitly)	Improved intent detection and response quality through contrastive and generative pre-training (numerical accuracy not explicitly stated)

1.3 Prior Art

Source/Title	"IntentGPT: Few-Shot Intent Discovery with Large Language Models," 2024
Task Solved	Few-shot intent classification
Approach/Model	Few-shot in-context learning leveraging GPT-4; semantic few-shot sampling
Data	Minimal labeled intent datasets enhanced by embedding similarity-based sampling
Metrics	Few-shot intent classification accuracy, efficiency in intent discovery (metrics not explicitly detailed)
Results	Effective few-shot intent discovery with minimal labeling; improved classification performance (exact numerical results not explicitly detailed)

2.1 Data Preparation/Description

Source dataset description:

[Bitext](): 26,872 question-answer pairs, 27 intents, 11 categories

[Customer-Service-for-LLM](): 2,700 question-answer pairs, 27 intents, 11 categories

EDA:

All queries are short, average length ≈ 8.69 words [Bitext], 8.6 words [CS-for-LLM]

Intent labels are pre-annotated, there are no missing values in both datasets.

Most common intents:

➔ contact_customer_service,complaint, check_invoice [Bitext]

➔ check_invoice, switch_account, edit_account [CS-for-LLM]

Relevant Fields & Labels for both datasets:

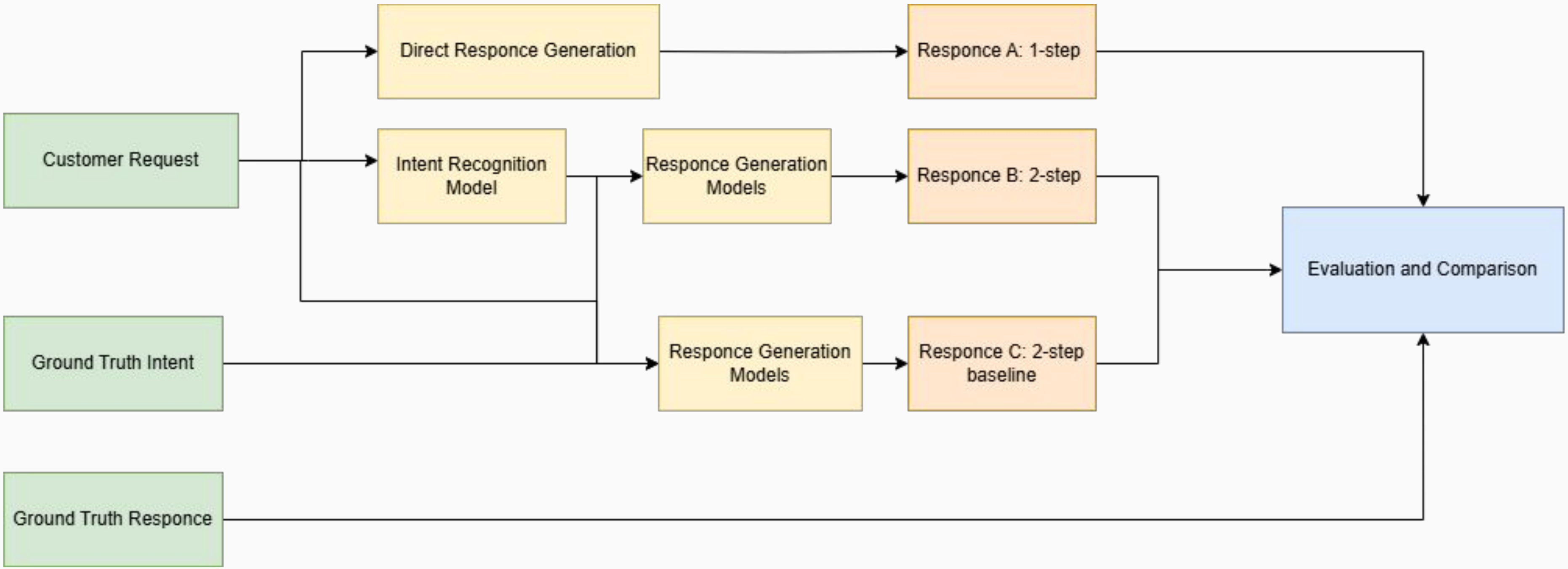
Input: instruction (user query): "I want to cancel my subscription"

Output: response (agent reply): "Sure, I can help with that. Please provide your account ID."

Intent: intent (pre-annotated intent label): cancel_service

2.2 Models and processing pipelines

Pipeline



2.2 Models and processing pipelines

Models configurations

Type	Intent Recognition Model	Response Generation Model
Single-step pretrained	None	T5
Single-step FT	None	Fine-Tuned T5
Two-step baseline	Ground Truth	T5
Two-step pre- trained	T5	T5
Two-step, partial FT	Fine-Tuned BERT	T5
Two-step, complete FT	Fine-Tuned BERT	Fine-Tuned T5

2.2 Models and processing pipelines

Data Splitting:

The dataset was split into **training (80%)** and **validation (20%)** sets using train_test_split from Scikit-learn.

Configuration Parameters: for both T5 and BERT-based models

- **Epochs:** 3
- **Learning Rate:** 5e-5
- **Batch Size:** 8
- **Weight Decay:** 0.01 Applied during T5 training to prevent overfitting.

Platform:

Local Machine with a compatible GPU (NVIDIA RTX 4060) for development and small-scale experiments. CUDA is used to leverage GPU acceleration.

2.3 Metrics

We used **both automatic and human evaluation metrics** for assessing generated chatbot responses.

- **Automatic Metrics:**
  BERTScore (Precision, Recall, F1)
  ROUGE (ROUGE-1, ROUGE-2, ROUGE-L F1 scores)
  BLEU (n-gram overlap for fluency and adequacy)

- **Human Evaluation Metrics:**
  Helpfulness
  Fluency
  Appropriateness

During the evaluation phase, metrics such as BERTScore, ROUGE, and BLEU are computed by comparing the generated responses to the reference responses in the test set, while human evaluation metrics are averaged over ratings of Helpfulness, Fluency, and Appropriateness provided by an LLM-based evaluator.

2.4 Code Organization

Repository

- Final_PPT
  - components
    - data_loader
    - evaluation
    - intent_recognition
    - response_generation
    - experiment_runner
  - config
  - .env
  - main
  - README_Final_PPT.md
  - requirements.txt

Loading the data
Cut the datasets to be the same size
Format according to the correct field names
Spit to train and test

Eval of the results Automatic Metrics + Human Evaluation Metrics

Intent Detection Fine-tuned BERT or pretrained T5

Response Generation: Pretrained or fine-tuned T5

Runs the pipeline according to the configuration

Main file that runs the experiment

metrics.json


```
{
    "bert_score_precision": 0.883563756942749,
    "bert_score_recall": 0.8590034246444702,
    "bert_score_f1": 0.8709178566932678,
    "rouge1_f": 0.390133741397222,
    "rouge2_f": 0.14120971398972432,
    "rougeL_f": 0.25122268401834436,
    "bleu": 0.05845409274338135,
    "Human_Helpfulness": 2.36,
    "Human_Fluency": 3.526666666666667,
    "Human_Appropriateness": 3.1333333333333333,
    "Human_Average": 3.006666666666667
}
```


human_scores.csv

query	intent	reference_response	generated_response	Helpfulness	Fluency	Appropriateness	avg_score

3.1 Intermediate/Baseline

Configuration	Intent Recognition Accuracy		Generation Metrics bert_score_f1 / rougeL_f / bleu / Human Evaluation Score	
	bitext	customer_service	bitext	customer_service
two_step_baseline	X	X	0.8116 0.0654 0.000000005779 1.9	0.8124 0.0602 0.0000000244 2.06
single_step_pretrained	X	X	0.801 0.0692 0.00000004137 1.9	0.8097 0.0729 0.0000003642 1.82
single_step_ft	X	X	0.868 0.2451 0.05537 2.66	0.8691 0.2468 0.05819 2.7
two_step_pretrained	0	0	0.7952 0.0716 0.00000002458 1.84	0.8028 0.0746 0.0000001003 1.83
two_step_partial_ft	0.9814	0.9944	0.8115 0.0652 0.0000000543 1.93	0.8123 0.06 0.000000002398 1.99
two_step_complete_ft	0.9851	0.9944	0.8692 0.2464 0.05851 2.76	0.8709 0.2512 0.05845 3.01

3.1 Intermediate/Baseline

- Baseline: **single_step_pretrained (no fine-tuning, no intent)**
- Main model: **two_step_complete_ft**
- Two-step models show consistent improvements in all metrics
- ***Fine-tuned two-step model (complete) performs best overall***

➔ Our main model (two_step_complete_ft) outperforms all baselines on both datasets and across all metrics.
It demonstrates that **conditioning on intent** and **full fine-tuning** are essential for high-quality customer support responses.

3.2 Main Results and conclusion

Model	Intent Accuracy	BLEU	Human Eval	Conclusion
single_step_pretrained	X	0.8097	1.82	Baseline
two_step_pretrained	0	0.8028	1.83	No FT
two_step_complete_ft	0.9944	0.8709	3.01	Best overall

✔ two_step_complete_ft achieves the best results across all metrics
✔ Fine-tuning both stages boosts BLEU, BERTScore, and ROUGE-L
✔ Two-step (intent-aware) models outperform single-step baselines
✔ Findings confirm: intent conditioning improves response quality

➔ We tested how intent conditioning and fine-tuning affect response quality. Results show that **fully fine-tuned, intent-aware models** generate more fluent and relevant answers.

4. Graphical Abstract