Conversational Agent

Final Presentation for MSAI 337 from Group 5

Tianchang Li
Liqian Ma
Shubham Shahi
Srik Gorthy

Introduction

Multi-Domain Wizard-of-Oz dataset (MultiWOZ), a fully-labeled collection of human-human written conversations spanning over multiple domains and topics.

At a size of 10k dialogues, it is at least one order of magnitude larger than all previous annotated task-oriented corpora.

Introduction

We are building a conversational agent using the Multi-WOZ dataset

This aids in a building a hand-on experience in building and evaluating language models and the various techniques

01

Data

Data and EDA

02

Approaches

Possible solutions

03

Results and Analysis

Scores and Commentary

04

Next Steps

What else?

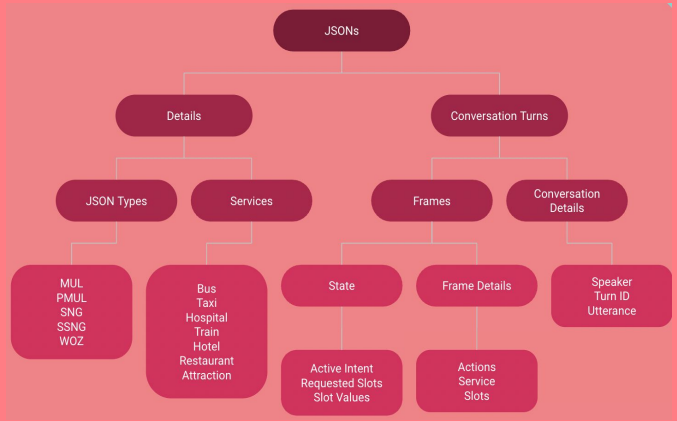
01

Data

Data and EDA

Data

Dialogue Types	# occurrences in all JSONs
PMUL	4332
MUL	2700
SNG	2341
WOZ	676
SSNG	388



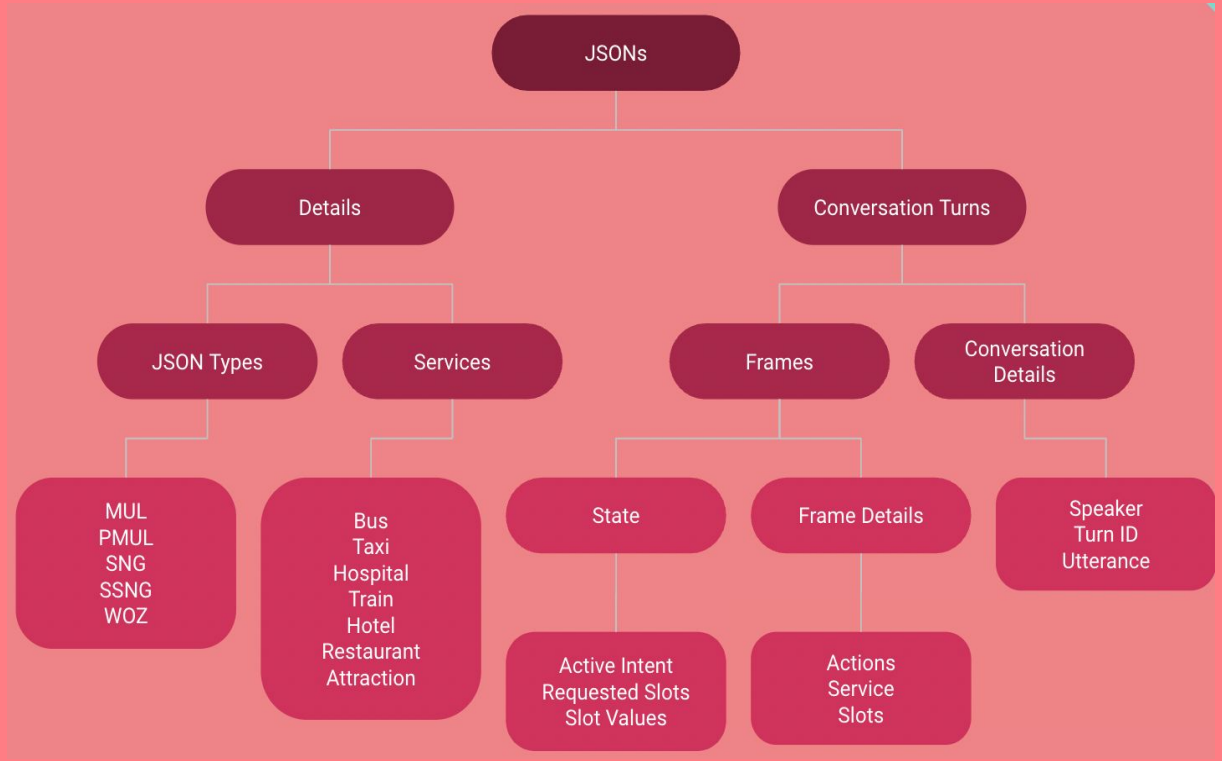
Service Types	# occurrences in all JSONs
Restaurant	4728
Hotel	4182
Train	3931
Attraction	3485
Taxi	872
Hospital	108
Bus	6

Dialogue Types

Data Structure

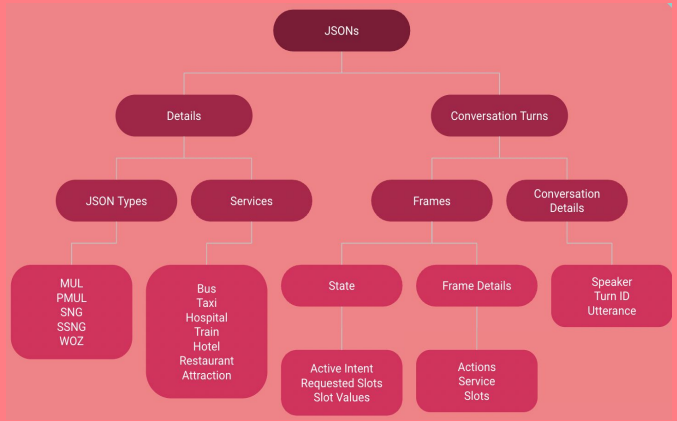
Service Types

Data


JSONs
Details
Conversation Turns
JSON Types
Services
Frames
Conversation Details
MUL
PMUL
SNG
SSNG
WOZ
Bus
Taxi
Hospital
Train
Hotel
Restaurant
Attraction
State
Frame Details
Speaker
Turn ID
Utterance
Active Intent
Requested Slots
Slot Values
Actions
Service
Slots

Data

Dialogue Types	# occurrences in all JSONs
PMUL	4332
MUL	2700
SNG	2341
WOZ	676
SSNG	388



Service Types	# occurrences in all JSONs
Restaurant	4728
Hotel	4182
Train	3931
Attraction	3485
Taxi	872
Hospital	108
Bus	6

Dialogue Types

Data Structure

Service Types

02

Approaches

Possible solutions we came up with
to build the agent

Approaches

01

Different Language Models

02

Different Decodings

03

Different Metrics

Approach 1 - Different Language Models

Encoder-Decoder:
T5, **XLNet**, RoBERTa

- Byte Pair encoding of a SentencePiece library
- Nearly 4 times more training data than GPT2 model, up to 146GB
- A presence of context for each of the words in the sentence
- Chose XLNet after comparing the performance of different models and the implementation difficulty

Decoder only:
GPT2, **GPT3**

- Achieved 0.13 bleu score in zero-shot test
- OOM on our machine
- 175 billion parameters and 570G training set
- Chose GPT3 after comparing the performance of different models and the implementation difficulty

Approach I - Insights

- **Training Epochs**
  - For the first 200 epochs, the model performance increases
  - After that, the model performance barely increased and started overfitting

- **Evaluation**
  - The [END] token can be a negative value for the BLEU score since all generated sentences do not contain it
  - For beam search, there are two spaces before the [END] token so that the next generated sentence is deprived of the first word

Approach II - Different Decodings

- **Results**
  - Beam Search performed well for all our models
  - Greedy gave us least ideal results

- **Insights and Commentary**
  - As **Greedy** decoding generates next word using maximum probability, the inaccuracies got aggregated at each word, giving us lesser performance
  - **Beam Search** improved the performance through better control between variance and bias of predictions, allowing deviation among the few best options(we set 5)
  - **Top-p Sampling** increased the variance of prediction and decreased the bias at the same time(we set p=0.9, allowing deviations among words accounting for first 90% of probabilities)

Approach III - Different Metrics

BLEU

- BLEU score measures the precision
- This represents how much the tokens in the model generated output text appeared in the human reference text
- BLEU score has a brevity penalty term and computes the n-gram match for several size of n-grams

ROUGE

- ROUGE measure has ROUGE-n Precision and ROUGE-n Recall statistics
- We chose the ROUGE-n recall for this project
- This represents how many tokens in the human references appeared in the model generated output text
- The ROUGE score doesn't have a brevity penalty and computes score for a given n

Approach III - Different Metrics

```
ref:   What is the phone number and address? [END]
pred:  is the address and phone number?
pred len = 32
ROUGE[95]:   0.286
ref:   How about portuguese food? [END]
pred:  about portuguese food?
pred len = 22
ROUGE[93]:   0.500
ref:   Thank you goodbye. [END]
pred:  you goodbye.
pred len = 12
ROUGE[97]:   0.333
```

Insights and Commentary

- By considering two scores, we hoped to obtain a well-rounded evaluation of both the precision and recall of the model and avoid type I/II errors
- All the models, other than the zero-shot, on average have a BLEU scores 20% higher than ROUGE. This indicates a higher precision than recall, which could result from an observation that predicted sequences are usually shorter than reference with the first word and [END] omitted
- We can add a F1-Metric using ROUGE-n precision and ROUGE-n recall as an alternative for comparison purpose

03

Results and Analysis

Scores and Commentary

Results - BLEU Scores

Baseline Model

Model/ Decoding	Zero Shot GPT-2	Fine-tuned	Fine-tuned (200 Epochs)	Mini GPT-3	XLNet
Greedy	0.081	0.39	0.54	0.44	0.56
Top-p	*negligible	0.41	0.57	0.43	0.58
Beam	0.074	0.43	0.61	0.49	0.64

* To be consistent, models are fine-tuned for 50 epochs except for zero shot and the GPT-2 fine-tuned for 200 epochs

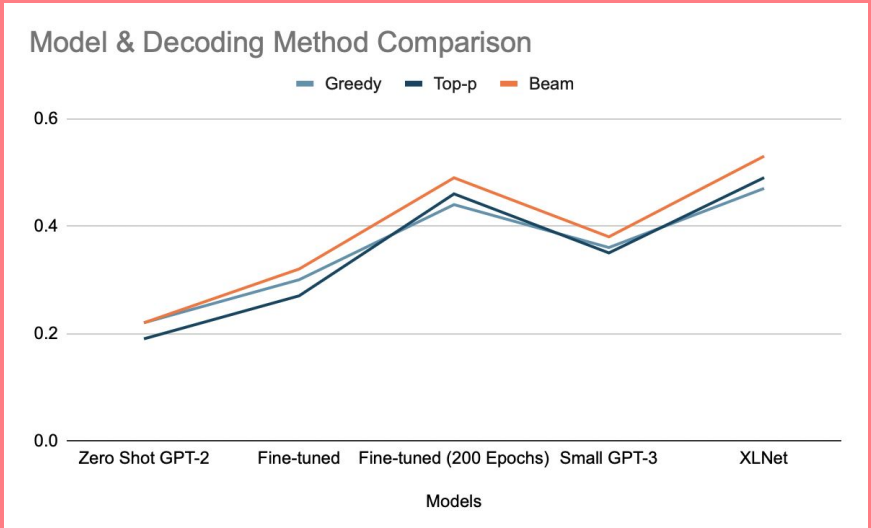
Results – ROUGE Scores

Baseline Model

Model/ Decoding	Zero Shot GPT-2	Fine-tuned	Fine-tuned (200 Epochs)	Mini GPT-3	XLNet
Greedy	0.22	0.30	0.44	0.36	0.47
Top-p	0.19	0.27	0.46	0.35	0.49
Beam	0.22	0.32	0.49	0.38	0.53

* To be consistent, models are fine-tuned for 50 epochs except for zero shot and the GPT-2 fine-tuned for 200 epochs

Results Analysis



ROUGE Scores

For Best Performance of Models,
Using model trained on larger dataset;
Building Fine-tuned model;
Fine tuning for more epochs if possible

As mentioned earlier,
Beam Search produces best results
as it balances variance and bias

The BLEU and ROUGE scores
remain consistent for
various models and methods
validating our conclusions

04

Next Steps

What else?

Methods which can be tried

01

Trying different encodings such as encoding days, phone numbers, addresses etc

02

Focusing on specific service types such as Restaurant, Hotel, Train and Attraction

03

Fine tune for optimal p in top-p sampling and # of beams in beam search

04

Data Augmentation by reversing the prompt and predictions sequences

Code is available at the following link

GITHUB

https://github.com/srikg-msai22/337NLP-G5-Final

Thanks!