Fine-tuning GPT-2 for Short Query Intent Classification

Jin Young Lee

June 11, 2025

Understanding Search Intent

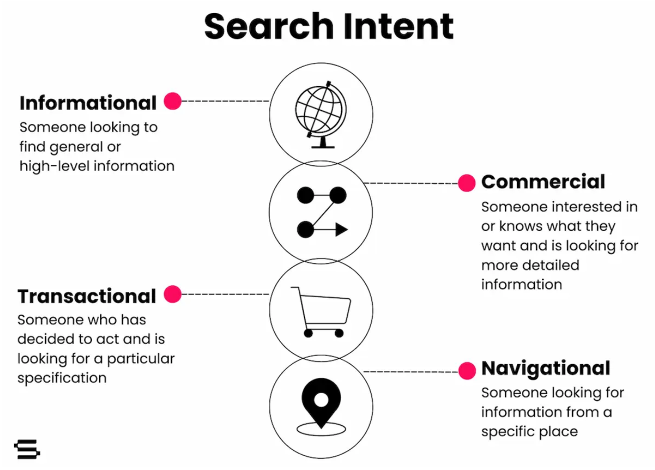

Figure: Different types of Search Intent

Motivation and Problem Statement

What is the problem? Why is it important?

- Short queries (`e.g., "weather?", "pizza nearby"`) are ambiguous
- Misclassified intents degrade user experience
- Accurate intent classification is crucial for:
  - Voice Assistants
  - Search Engines
- **Challenge:** Minimal context in short queries

Proposed Approach and Methodology

Category	Detail
Model	GPT-2 base model (768d hidden)
Classification Head	Custom linear layer
Fine-tuning Modes	Last-linear-layer, Full-model
Regularization	Dropout (0.3)
Optimizer	AdamW
Learning Rate	1e-3
Batch Size	8
Loss Function	Cross-entropy
Early Stopping	Based on dev accuracy

Amazon MASSIVE Dataset (EN-US Subset)

[View on Hugging Face]

Feature	**Description**
Total Languages	51 (Multilingual)
Subset Used	`en-US` only
Utterance Count	∼60,000 utterances (EN-US)
Intent Classes	60 distinct intent types
Domains	Music, Weather, Alarms, Smart Home, etc.
Utterance Length	Mix of short and long queries
Label Quality	Human-annotated, high quality
Source	Amazon Alexa / MASSIVE Dataset

MASSIVE Dataset: EN-US Subset Examples

ID	Utterance	Intent Label
1	wake me up at nine am on friday	alarm_set
2	set an alarm for two hours from now	alarm_set
5	stop	audio_volume_mute
9	make the lighting bit more warm here	iot_hue_lightchange
15	turn off the light in the bathroom	iot_hue_lightoff
22	dim the lights in the kitchen	iot_hue_lightdim
25	olly clean the flat	iot_cleaning
33	check when the show starts	calendar_query
34	i want to listen arijit singh song once again	play_music

Intent types include alarms, smart lighting, music playback, and calendar access.

Data Preprocessing Pipeline - Input Example

Preprocessing Steps for GPT-2:

- Tokenize input using GPT-2 tokenizer (BPE-based)
- Apply dynamic padding within batch
- Truncate to max model input length (e.g., 128)
- Use <|endoftext|> as padding token
- Track unique utterance ID for evaluation alignment

Example Utterance:

- **Text:** "play some jazz music"
- **Intent:** `music.play_song`

Tokenization Flow: From Text to Embeddings

Raw Utterance: `play some jazz music`

↓

BPE Tokens: `["play", " some", " jazz", " music"]`

↓

Token IDs: `[1824, 142, 3678, 920]`

↓

Attention Mask: `[1, 1, 1, 1]`

↓

Embedding Vectors: `[4 × 768-d] vectors`

Training Progress

Loss and Metrics Tracking

- Training loss decreases steadily
- Validation metrics show convergence
- No significant overfitting observed
- Full-model fine-tuning shows better convergence

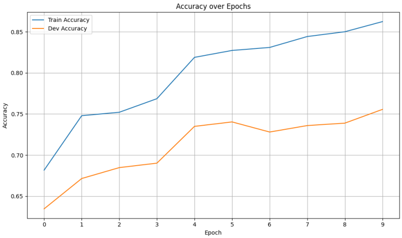

Figure: Example: Accuracy over Epochs

Training and Development Loss

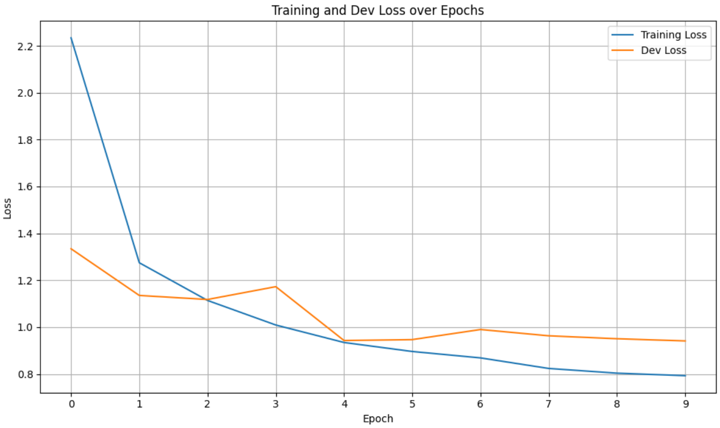

Figure: Training and Development Loss over Epochs

Accuracy over Epochs

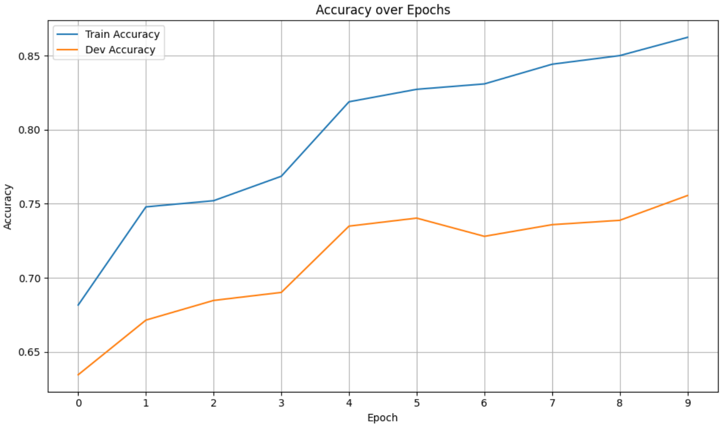

Figure: Accuracy Performance over Epochs (Last-Linear-Layer)

Accuracy Comparison: Last-Linear-Layer vs. Full-Model

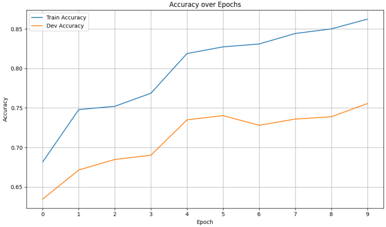

Figure: Last-Linear-Layer Accuracy

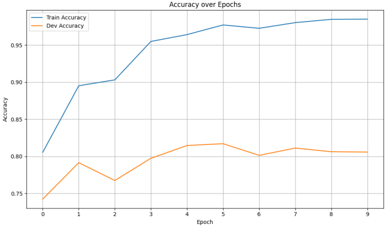

Figure: Full-Model Accuracy

F1 Score Comparison: Last-Linear-Layer vs. Full-Model

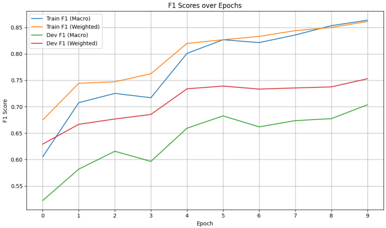

Figure: Last-Linear-Layer F1 Scores

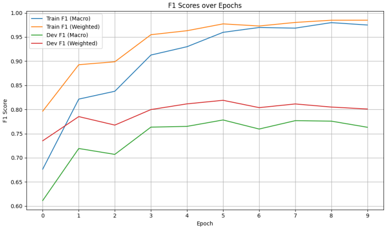

Figure: Full-Model F1 Scores

Comparison to Official MASSIVE Benchmark (EN-US)

Intent Accuracy on EN-US Subset:

`FitzGerald et al., 2022 (arXiv:2204.08582)`

Model	Type	Accuracy (%)
GPT-2	Decoder-only (monolingual)	**80.1**
mT5 Enc Full	Encoder-decoder (multilingual)	89.0 ± 1.1
mT5 T2T Full	Encoder-decoder (multilingual)	87.9 ± 1.2
XLM-R Full	Encoder-only (multilingual)	88.3 ± 1.2

Interpretation:

- Our GPT-2 model achieves **competitive performance** without cross-lingual supervision.
- Models in the paper use **larger pretraining corpora**, **multilingual tokens**, and more parameters.

$\rightarrow$ *For an English-only GPT-2 baseline, 80% accuracy is strong given the task complexity.*

Ours vs. Original MASSIVE Benchmark

Original MASSIVE Benchmark (FitzGerald et al., 2022)

`arXiv:2204.08582`

- mT5 / XLM-R models trained using:
  - `p3dn.24xlarge` (8x V100 GPUs) for 3–5 days
  - `g4dn.metal` (8x T4 GPUs) for mT5 Encoder
  - Extensive hyperparameter tuning on multilingual data

Our Setup (GPT-2):

- Single `RTX` 3080 GPU
- Training time: **3 hours total**
- No multilingual pretraining or zero-shot setup

Conclusion: Our monolingual GPT-2 model achieves competitive performance with only a fraction of the computational cost.