

Structured Sentiment Analysis using In-Context Learning

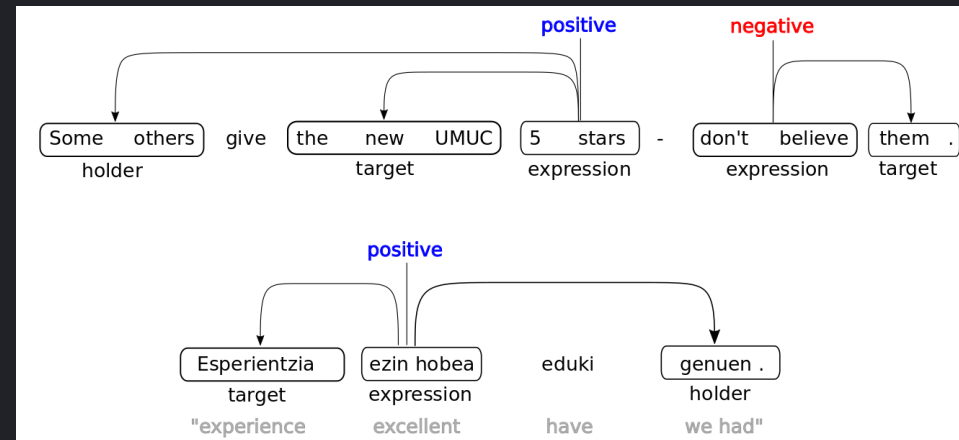
Martin Popovski

Who am I?

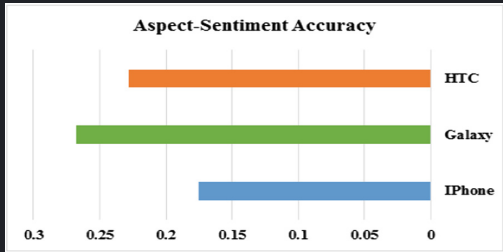
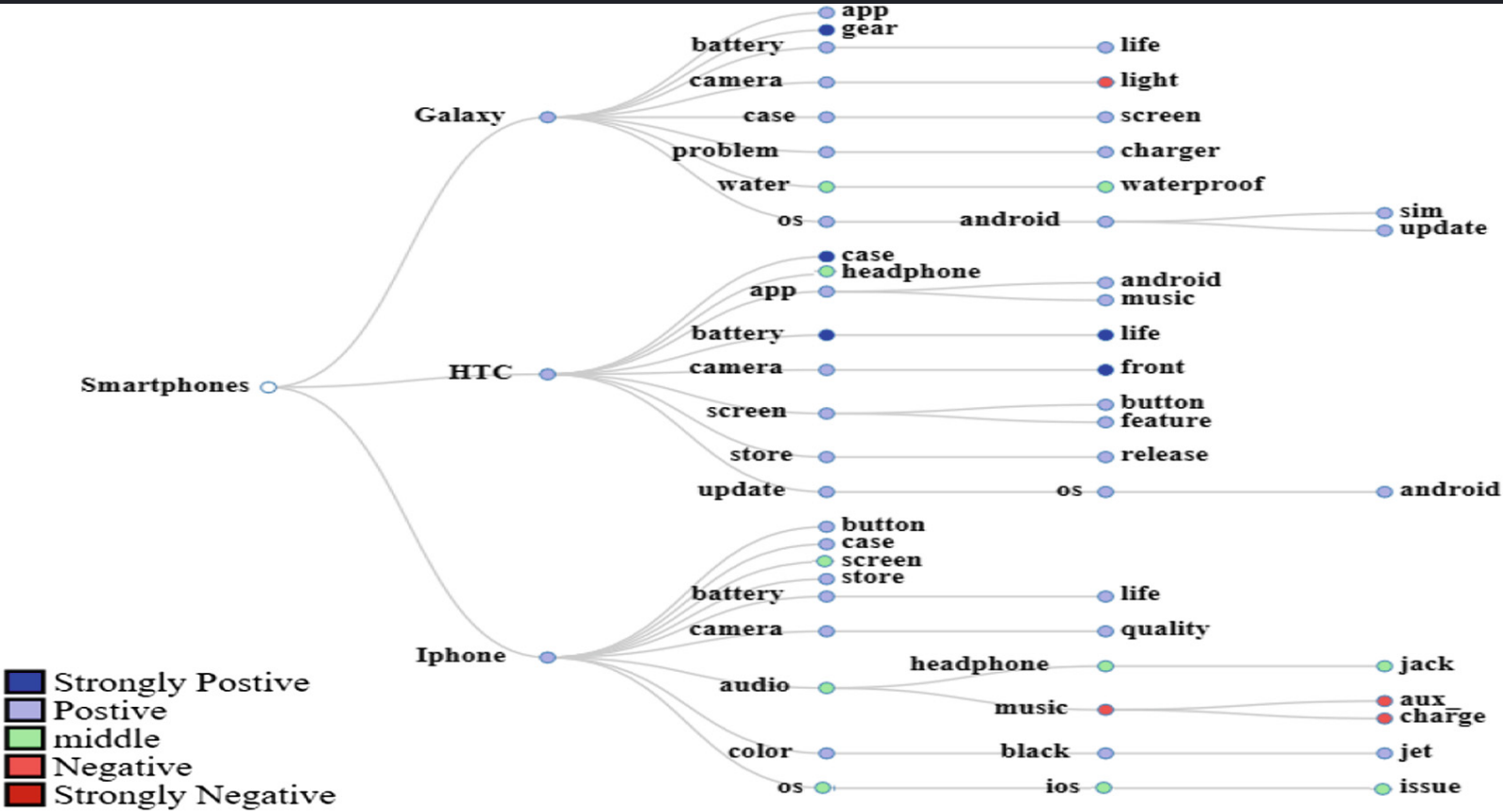
- Education
 - FINKI
 - Generation 2018
 - Major Computer Science
 - Took NLP course
- Professional experience
 - 2 years at Netcetera
 - Machine Learning Engineer
 - Mainly working with Large Language Models

The task

- Extract opinion tuples from a sentence
 - opinion holder
 - opinion target
 - opinion expression
 - opinion polarity



Example use case



The data

	sentences		holders			targets			expressions			polarity		
	#	avg.	#	avg.	max	#	avg.	max	#	avg.	max	+	neu	-
MultiBooked _{EU}	1,520	10.6	296	1.1	6	1,760	1.4	9	2,319	2.2	10	1,940	0	379
MultiBooked _{CA}	1,676	15.2	237	1.1	7	2,350	2.4	18	2,770	2.6	19	1,743	0	1,027
OpeNER _{EN}	2,492	14.8	413	1.0	3	3,843	1.8	21	4,149	2.4	21	2,981	0	1,168
OpeNER _{ES}	2,054	17.4	225	1.0	2	3,960	2.2	12	4,386	2.2	15	3,557	0	829
MPQA	10,048	23.3	2,265	2.7	40	2,437	6.3	50	2,794	2.0	14	1,082	465	1,059
DS _{Unis}	2,803	20.0	94	1.2	4	1,601	1.2	6	1,082	1.9	9	612	186	805
NoReC _{Fine}	11,437	16.9	1,128	1.0	12	8,923	2.0	35	11,115	5.0	40	7,547	0	3,557

Example sample

```
{
  "sent_id": "ula/116CUL032-6",
  "text": "As a provider of youth services , MCCOY , Inc. is here to support your valuable efforts to develop young people .",
  "opinions": [
    {
      "Source": [
        ["MCCOY , Inc."],
        ["34:46"]
      ],
      "Target": [
        ["efforts"],
        ["80:87"]
      ],
      "Polar_expression": [
        ["support"],
        ["58:65"]
      ],
      "Polarity": "Neutral",
      "Intensity": "Average"
    }
  ]
}
```

Traditional Approaches vs In-Context Learning

Traditional Approaches

- Fine tune a pre-trained model on the specific task
 - e.g. BERT, RoBERTa, XLM-RoBERTa etc.
- Requires a lot of data
- Computationally expensive to train (fine tune)
- Computationally efficient to use

In-Context Learning

- Large language models are fine tuned on general instruction following tasks
 - e.g. GPT-3.5 Turbo, Llama 2 Chat, WizardLM, Vicuna, Orca, Platypus etc.
- Requires a lot less data
- Task specific fine tuning is not required
 - Although possible
- Computationally inefficient to use
 - Depending on the model size

What LLM should I use?

- Cloud based
 - OpenAI
 - GPT-3.5
 - GPT-4
- Locally hosted
 - Llama 2
 - Mistral
 - Falcon
 - Yi
 - Tigerbot
 - Many more

Hardware requirements for local LLMs

Quantization

- Floating point precision
 - 16 bit - default precision for most LLMs
 - 8 bit quantization - ~2x memory reduction
 - 4 bit quantization - ~4x memory reduction
- Quantization formats
 - GGML and GGUF
 - CPU+GPU inference
 - GPTQ and AWQ
 - GPU inference

Model	Original Size	Quantized Size (4-bit)
7B	13 GB	3.9 GB
13B	24 GB	7.8 GB
33B	60 GB	19.5 GB
65B	120 GB	38.5 GB

Hardware examples

- 8 GB VRAM
 - GTX 1070, GTX 1080, RTX 2060 Super, RTX 2070, RTX 2080, RTX 3070
 - 7B at 4 bit quantization
- 12 GB VRAM
 - RTX 3060, RTX 3080
 - 13B at 4 bit quantization
- 24 GB VRAM
 - RTX 3090, RTX 4090
 - 13B at 8 bit quantization
 - ~30B at 4 bit quantization
- 48 GB VRAM
 - 2 * RTX 3090 / RTX 4090
 - ~30B at 8 bit quantization
 - ~70B at 4 bit quantization

Choosing the right LLM

- https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Open LLM Leaderboard

The Open LLM Leaderboard aims to track, rank and evaluate open LLMs and chatbots.

Submit a model for automated evaluation on the GPU cluster on the "Submit" page! The leaderboard's backend runs the great [Eleuther AI Language Model Evaluation Harness](#) - read more details in the "About" page!

[LLM Benchmark](#) [Metrics through time](#) [About](#) [Submit here!](#)

Select columns to show

☒ Average

☒ ARC

☒ HellaSwag

☒ MMLU

☒ TruthfulQA

☒ Winogrande

☒ GSM8K

☐ Type

☐ Architecture

☐ Precision

☐ Hub License

☒ #Params (B)

☐ Hub

☐ Available on the hub

☐ Model sha

☐ Show gated/private/deleted models

Model types

☒ pretrained ☒ fine-tuned ☒ instruction-tuned ☒ RL-tuned ☒ ?

Precision

☒ float16 ☒ bfloat16 ☒ 8bit ☒ 4bit ☒ GPTQ ☒ ?

Model sizes (in billions of parameters)

☐ ? ☒ ~1.5 ☒ ~3 ☒ ~7 ☐ ~13 ☐ ~35 ☐ ~60 ☐ 70+

T	Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	#Params (B)
	fblgit/una-cybertron-7b-v2-bf16	69.67	68.26	85.85	63.23	64.63	80.98	55.04	7.24
	fblgit/una-cybertron-7b-v1-fp16	69.49	68.43	85.42	63.34	63.28	81.37	55.12	7.24
	Q-bert/Optimus-7B	69.09	65.44	85.41	63.61	55.79	78.77	65.5	7.24
	chargoddard/loyal-piano-m7-cdpo	69.08	67.15	85.39	64.52	61.53	79.4	56.48	7.24
	chargoddard/loyal-piano-m7-cdpo	69	67.06	85.42	64.54	61.54	79.08	56.33	7.24
	chargoddard/loyal-piano-m7	68.67	66.72	85.03	64.43	60.03	79.08	56.71	7.24
	Intel/neural-chat-7b-v3-2	68.29	67.49	83.92	63.55	59.68	79.95	55.12	7
	mlabonne/NeuralHermes-2.5-Mistral-7B	68.22	66.55	84.9	63.32	54.93	78.3	61.33	7.24
	Weyaxi/OpenHermes-2.5-neural-chat-7b-v3-1-7B	67.84	66.55	84.47	63.34	61.22	78.37	53.07	7.24
	perlthoughts/Chupacabra-7B	67.76	66.81	83.52	62.68	52.31	79.08	62.17	7.24
	Q-bert/Bumblebee-7B	67.73	63.4	84.16	64	50.96	78.22	65.66	7.24
	mrfakename/NeuralOrca-7B-v1	67.64	65.27	85.07	63.68	54.58	78.77	58.45	7.24

What to look out for

- Production release or test release
- What is it fine tuned for?
 - General instruction following
 - Conversing
 - Code generation
 - Role playing
- What languages does it support?
- Context size
- Licence

Note the prompt template

Prompt

The model is very good, works well on almost any prompt but ChatML format and Alpaca System gets the best

```
<|im_start|>system
- You are a helpful assistant chatbot trained by MosaicML.
- You answer questions.
- You are excited to be able to help the user, but will refuse to do anything that
- You are more than just an information source, you are also able to write poetry
<|im_start|>user
Explain QKV<|im_end|>
<|im_start|>assistant
```

```
### Assistant: I am StableVicuna, a large language model created by CarperAI. I am
### Human: Explain QKV
### Assistant:
```


```
[Round <|round|>]
问: Explain QKV
答:
```

```
[Round <|round|>]
Question: Explain QKV
Answer:
```

```
Question: Explain QKV
Answer:
```


Find a quantized version of the model

- <https://huggingface.co/TheBloke>



1181 11

Tom Jobbins PRO

TheBloke

Follow


10327 followers · 8 following

TheBlokeAI TheBloke

AI & ML interests

LLM: quantisation, fine tuning

Organizations



Collections 1

Recent models: last 100 repos, sorted by creation date

The last 100 repos I have created. Sorted by creation date descending, so the most r...

- TheBloke/SUS-Chat-34B-GPTQ
Text Generation · Updated about 4 hours ago · 1
- TheBloke/SUS-Chat-34B-AWQ
Text Generation · Updated about 6 hours ago · 2
- TheBloke/SUS-Chat-34B-GGUF
Text Generation · Updated about 7 hours ago · 7
- TheBloke/sabia-7B-GGUF

Models 2899

Sort: Recently updated

TheBloke/SUS-Chat-34B-GPTQ Text Generation · Updated about 4 hours ago · 1	TheBloke/SUS-Chat-34B-AWQ Text Generation · Updated about 6 hours ago · 2
TheBloke/SUS-Chat-34B-GGUF Text Generation · Updated about 7 hours ago · 7	TheBloke/sabia-7B-GPTQ Text Generation · Updated about 7 hours ago · 2
TheBloke/sabia-7B-AWQ Text Generation · Updated about 7 hours ago · 1	TheBloke/sabia-7B-GGUF Updated about 8 hours ago · 1 · 1
TheBloke/OpenOrca-Zephyr-7B-GPTQ Text Generation · Updated 1 day ago · 21 · 1	TheBloke/OpenOrca-Zephyr-7B-AWQ Text Generation · Updated 1 day ago · 34
TheBloke/OpenOrca-Zephyr-7B-GGUF Updated 1 day ago · 5 · 5	TheBloke/Poro-34B-GPTQ Text Generation · Updated 1 day ago · 1 · 2

Expand 2899 models

How to host the model locally?

- oobabooga's Text generation web UI
- ollama
- Text Generation Inference
- transformers

Meanwhile GPT
users be like



What I used

- RTX 3070 8 GB VRAM
- Orca Mini V3
 - Based on Llama 2 7B
 - GPTQ 4 bit 128 group size with act order
- oobabooga's Text generation web UI
 - ExLlama backend

Example sample again

```
{
  "sent_id": "ula/116CUL032-6",
  "text": "As a provider of youth services , MCCOY , Inc. is here to support your valuable efforts to develop young people .",
  "opinions": [
    {
      "Source": [
        ["MCCOY , Inc."],
        ["34:46"]
      ],
      "Target": [
        ["efforts"],
        ["80:87"]
      ],
      "Polar_expression": [
        ["support"],
        ["58:65"]
      ],
      "Polarity": "Neutral",
      "Intensity": "Average"
    }
  ]
}
```

First attempt

- Ideas
 - Provide the LLM with few shot examples from the training data
 - Define the expected output JSON schema to the LLM
 - Give the LLM the test data sentence as input
 - Parse the output of the LLM as a JSON
- Problems
 - The smaller 7B model is not capable of handling the complexity of the JSON
 - It doesn't conform to the JSON schema
 - The model doesn't always give an exact substring from the sentence
 - It corrects misspellings and spacings even when told not to
 - LLMs have no concept of character counts and offsets
 - They work with tokens

Second attempt

- Ideas
 - The substring position can be calculated afterwards deterministically
 - Use fuzzy search using Levinshtein distance to find the closest match substring in the sentence
 - Most probable match under a dynamic distance threshold
 - Simplify the JSON schema

Simplified Option Example

```
{  
  "Source": "MCCOY , Inc.",  
  "Target": "efforts",  
  "Polar_expression": "support",  
  "Polarity": "Neutral",  
  "Intensity": "Average"  
}
```


Simplified Option Schema

```
{
  "$defs": {
    "Intensity": {
      "enum": ["Average", "Strong", "Standard", "Weak", "Slight"],
      "title": "Intensity",
      "type": "string"
    },
    "Polarity": {
      "enum": ["Neutral", "Positive", "Negative"],
      "title": "Polarity",
      "type": "string"
    }
  },
  "properties": {
    "Source": {
      "anyOf": [{"type": "string"}, {"type": "null"}],
      "title": "Source"
    },
    "Target": {
      "anyOf": [{"type": "string"}, {"type": "null"}],
      "title": "Target"
    },
    "Polar_expression": {
      "title": "Polar Expression",
      "type": "string"
    },
    "Polarity": {
      "$ref": "#/$defs/Polarity"
    },
    "Intensity": {
      "anyOf": [{"$ref": "#/$defs/Intensity"}, {"type": "null"}]
    }
  },
  "required": ["Source", "Target", "Polar_expression", "Polarity", "Intensity"],
  "title": "SimpleOpinion",
  "type": "object"
}
```

Prompt Template

```
### System:
You are an AI assistant that follows instruction extremely well. Help as much as you can.

### User:
The task is to predict all structured sentiment graphs in a text (see the examples below).
We can formalize this as finding all the opinion tuples  $O = O_1, \dots, O_n$  in a text.
Each opinion  $O_i$  is a tuple  $(h, t, e, p)$  where  $h$  is a holder who expresses a polarity  $p$  towards a target  $t$  through a sentiment expression  $e$ ,
implicitly defining the relationships between the elements of a sentiment graph.
The response should be a representation of the opinions and must always be in the form of a json list of objects.
Each object must follow the following json schema:
{schema}
The response fields Source and Target may be null or a string.
The fields Source, Target and Polar_expression must be exact substrings from the input text,
they must not be altered in any way, no spelling corrections, no formatting corrections,
no skipping or adding words, the field must be an exact character for character substring in the input text.
The field Polarity must be one of the following: "Neutral", "Positive", "Negative".
Intensity must be one of the following: null, "Average", "Strong", "Standard", "Weak", "Slight".
If there are no opinions in the text, the response should be an empty list.

{examples}
### User:
{input}

### Assistant:
```

Choice of few shot examples

- The examples should be as diverse as possible and cover as many edge cases as possible
- 20 examples fit in the context size of 4096 tokens
 - Shuffle train dataset
 - 1 example with 0 opinions
 - 9 examples with 1+ opinion
 - 10 examples with 2+ opinions
- Possible improvements for example selection?
 - Store the sentences in a vector database and select the most similar examples to the input sentence

How to handle LLM invalid responses?

- Use a retry strategy
 - Retry 5 times with the same prompt, if all 5 responses are invalid, predict no opinions

Extra reading resources

- <https://www.reddit.com/r/LocalLLaMA/wiki/index/>
 - New models, discussions, hardware, hosting, etc.
- <https://python.langchain.com/docs/>
 - LangChain LLM framework
- <https://docs.llamaindex.ai/>
 - LlamaIndex LLM framework specialized for RAGs
- <https://www.deeplearning.ai/short-courses/>
 - Good free courses for LLMs and LangChain
- https://github.com/martinkozle/NLP-semeval22_structured_sentiment/tree/master/llm_solution
 - My code for this project
- <https://github.com/martinkozle/NLP-guest-presentation>
 - This presentation

Discussion