**PragmaticLM**

# **Goal**

To bridge the understanding gap between humans and computers by building a more nuanced and context aware NLP model. The model would enhance the understanding of LMs beyond lexical and semantic meanings, and aim at achieving pragmatic understanding of user prompts.

The aim of this paper is to make the ultimate user-intent understanding model which can be used for any AI inference task, for better human-AI collaboration. I aim to make this the go-to pipeline to be used with any other AI model, as the sole purpose of this model is to understand what the user wants the model to do, and direct the main model to do that in simple prompting suitable for the particular model.

This will be a transition from the literal text interpretation to genuine language comprehension, beyond explicit statements.

It should have;

1. Inherent flexibility
2. Cultural awareness
3. Theory of Mind capability (False-belief understanding)
4. Chain of Thought (CoT) compatibility
5. Generator + relationship finder

Since this concerns user interaction and predominantly user input, I am building a model which can take in a prompt and restructure it in the optimal format for the main target model for further generation or any other task. It will try to catch the intention and context, follow a brief CoT procedure to verify the understanding, and then generates the new computer-comprehensible prompt for optimal output.

**Retrieval:** A retrieval system can also be integrated for fact checking and context fetching, for low confidence tokens and contexts.

# **Relevance**

PragmaticLM is relevant, because it:

1. **Addresses a real challenge in NLP**User inputs are often ambiguous or overloaded with context, and not optimal for generative AI models,
2. **Combines Retrieval, reformulation, and pragmatics (and maybe Knowledge graphs)**Acts as the context – mediator.
3. **Can bridge the communication gap between humans and intelligent models**It can do so by understanding the lexical, semantical, and pragmatic meaning of the user inputs and by capturing the user intent from the prompts.

Pragmatic understanding is the Fifth and highest phase in NLP development.

Levels of linguistic understandings;

1. Lexical – *(words*, *vocabulary)*,
2. Syntactic - *(sentence level understanding)*,
3. Semantic - *(comprehending meaning)*,
4. Discourse - *(connect sentences and collective meaning)*
5. Pragmatics – *(understanding of intent and goals)*

Current LLMs demonstrate remarkable capabilities in semantic processing, **but falter at pragmatic reasoning tasks.**

***Pragmatics*** *is essentially beyond surface level text interpretation.*

# **Current Challenges and Bottlenecks**

1. **Lack of sufficient annotated data**Context aware annotations
2. **Layer Depth mismatch**not sure which layers to be trained for pragmatics
3. **Unidentified training methods**PO is the best suited so far, still room for improvement

# **Methodology**

The model would be fine-tuned and built on top of existing language models.

PragmaticLM will be initially built to handle the following the tasks;

1. Handle indirect requests – *(asking without explicitly asking)*
2. Use an extended context window besides just the input to understand the intent better

## **Existing architectures**

**Casual LMs**

These are essentially next word prediction models. They use only decoder structures. These can be trained to follow CoT and reasoning steps.   
***Relevance to PragmaticLM:*** *These models can be trained on sufficient data and in a relevant way to build pragmatic understanding.*

**Masked LMs**

These are BERT like models, trained to predict missing or "masked" words within a piece of text, helping them learn contextual relationships between words.

***Relevance to PraagmaticLM:*** *These can be used to find relevance and relationship between input prompt and generated restructured prompt.*

**Autoregressive LMs**

These models are good at focusing on **semantics** and **disclosure** for next token prediction.

**Final selection**

**T5 Models** are best suited for PragmaticLM, as it is a sequence-to-sequence model, an Encoder-Decoder based model. This architecture is relevant for;

* It is best suited for generating corresponding output text as restructured prompts.
* The encoder layers facilitate a self-attention mechanism for input tokens, which helps in identifying input tokens which attend to each other.
* The decoder layers facilitate a cross attention mechanism for input-output tokens.
* This contributes towards the goal of PragmaticLM to get an understanding of the inputs before generating the relevant output.

# **T5 Architecture**

**T5 – Text to text Transfer Transformers**

* Encoder – decoder model (E-D)
* Sequence – to – sequence model (seq2seq)

## **Encoder**

**A high-level understanding of the workflow;**  
It takes the entire input sequence and compresses it into fixed length representation, the context vector or hidden state. It then **captures** the meaning of the input sequence using self-attention mechanism.

## **Decoder**

**A high-level understanding of the workflow;**  
It takes the encoded representation and generate s an output sequence, step by step, one token at a time. It uses the previously generated tokens as input for next step.

**T5 can be extensively fine-tuned to;**

1. Understand the intent of the user by self-attention
2. Use cross attention to;
   1. follow a CoT procedure to reason, and
   2. generate a restructured and relevant version of the prompt

Its output can then be passed on to the main model in the pipeline for the main task execution.

# **T5 Customization for PragmaticLM**

A new task configuration will be defined for prompt restructuring. Also, additional layers could be added to the model for pragmatic understanding, and can be trained on high quality data.

**Recommendations for Improvement**

1. **Clarify Evaluation Metrics:**  
   Define clear, quantifiable metrics for assessing pragmatic understanding. Consider leveraging human-in-the-loop evaluations and benchmark tasks specifically designed to test indirect requests and contextual nuance.
2. **Data Strategy:**  
   Develop a robust plan for obtaining high-quality, context-aware annotations. This might include crowdsourcing, leveraging existing datasets, or designing new tasks that capture cultural and intent nuances.
3. **Pilot Study:**  
   Start with a pilot implementation focused on a narrow domain. This will allow you to test core components (e.g., prompt restructuring and retrieval) before scaling the model to handle more generalized tasks.
4. **Interdisciplinary Collaboration:**  
   Engage experts in linguistics, cognitive science, and human-computer interaction to help frame the abstract elements (like cultural awareness and theory of mind) into measurable components.
5. **Comparative Analysis:**  
   Benchmark PragmaticLM against existing meta-prompting techniques and models that attempt to understand indirect language. This will not only validate your approach but also highlight its unique contributions.