




Misconception	Reality
Silver bullet for all Natural Language Processing (NLP) tasks	LLMs are powerful tools, but not a one-size-fits-all solution.
Solution	
Evaluate traditional NLP techniques or architectures (such as BERT) that may be more suitable for specific problems, especially those that need analysis (not generation)	




Misconception	Reality
One-Shot Prompting	Despite impressive research and academic examples, LLMs typically require multiple examples in a prompt to deliver meaningful results for a complex task
Solution	
Try few shot prompting with several specific relevant examples to guide the model towards desired output	




Misconception	Reality
Perfect Intent Understanding	LLMs may not always grasp user intent or context. Additional information or clarification might be necessary for accurate responses which could affect solution architecture
Solution	
Use prompt engineering techniques like Chain-of-thought (CoT), Reason-Action (ReAct) or Tree-of-thought (ToT) to get better results	



Misconception	Reality
Minimal Data Fine-Tuning	Fine-tuning for specific tasks often demands substantial task-specific data for optimal performance. An under-trained LLM could lead to performance degradation with time when data encountered in production is no longer similar to that used in fine-tuning
Solution	
The choice between Retrieval Augmented Generation (RAG) and fine-tuning should be carefully made. In most cases, when ground truth data is dynamic, RAG is the better solution	



Misconception	Reality
Bigger is Better	Large foundation models are trained to provide acceptable results to the average retail users. However, for enterprise applications, smaller and niche models can not only save money but also provide higher-quality output
Solution	
Experiment with smaller, niche models. Consider how these models can be used in tandem with larger models, for multi-step LLM generation. Fine-tune smaller models when required	



Misconception	Reality
Retrieval Augmented Generation (RAG) is the solution to all search problems	Not all RAG implementations deliver a quality search user-experience
Solution	
<ol style="list-style-type: none"> 1. Experiment with different embedding models to determine which provides the best results on your searchable data assets. 2. Refine how data assets are broken into 'chunks' for the vector database. 3. Explore using tag extract and context analysis to provide additional metadata during the pre-processing data assets fed into the vector database. 	