Loris

THE CX LEADER'S PRIMER

Navigating Generative AI without the Jargon

EMBRACING GENERATIVE AI:

Why CX Leaders Can't Ignore This Development

by Eugene Mandel

If you are a leader in a CX organization, you probably have seen a lot of demos of AI capabilities that seem to really "get" the language of customer support in the past year – from more intuitive chatbots to tools that help agents write their messages according to brand tone. You have heard that what drives this magic is ChatGPT and Generative AI, and that this is about to change EVERYTHING.

This guide will explain the main concepts behind these developments without technical jargon.

After reading it, you as a CX leader will understand:

1. What these concepts are
2. Why they matter
3. How they are different from what existed before
4. What it means to you

Generative AI is not magic. It is actually built utilizing a particular kind of innovative Machine Learning models called Large Language Models (LLMs).

LLMs represent a jump in what Natural Language AI can do for Customer Support. They are not going to replace all humans in CX any time soon, but they will transform many aspects of CX operations, both customer-facing and internal. Ignoring this development is not an option for a CX leader – LLMs are not a "fad".

LLMs are the third generation of Natural Language Processing (NLP), technology that Customer Support teams have been applying for decades.

First Generation of NLP – Keyword Based Search

Keyword based search allows finding documents and messages that contain particular keywords. The major issue with this technology is that you'd have to already know all the keywords that you are searching for. If you want to tag customer support conversation transcripts with sentiment, you would have to add all the keywords that signal positive, neutral and negative sentiment. Human language offers such a variety of expressing every emotion that this is a losing battle.

Second Generation of NLP – Machine Learning

Machine Learning (ML) solves the problem of explicitly managing the exact keywords that express a particular concept.

An ML model is a software program created to perform a particular task, but instead of programmers writing code that specifies how to perform the task, they are training the model to perform the task. Training is done by showing the machine a set of example problems with their solutions. It is similar to showing a picture of a cat to a toddler and saying, "look, this is a kitty", then a picture of a dog and saying, "look, this is not a cat, this is a doggy". Unlike a human toddler, a model would need hundreds, if not thousands, of pictures labeled "cat" and "not cat" to learn to perform the task of deciding if there is a cat in a picture.

Third Generation of NLP – Generative AI and Large Language Models

An LLM is a Machine Learning model. Each ML model is trained for a specific task.

The task that LLMs are trained for is to predict next words, given a piece of text. For example, given the text “We are finishing each other’s”, a model can decide that a plausible next word is “sandwiches” or “sentences”. This explains the second “L” in “LLM” – it stands for “Language”.

All ML models predict something, but predicting the next plausible word in a text is different from predicting the likelihood of a picture containing a cat. By adding the next predicted word to the text and then predicting the next word again (and again, and again...), the model writes (or generates) new text that never existed before.

{ The best thing about AI is its ability to,

The best thing about AI is its ability to create,

The best thing about AI is its ability to create worlds,

The best thing about AI is its ability to create worlds that,

The best thing about AI is its ability to create worlds that are,

The best thing about AI is its ability to create worlds that are both,

The best thing about AI is its ability to create worlds that are both exciting, }

This is why LLMs are also called “generative models”, to differentiate them from models that make predictions and decisions about existing documents and images. In our cat image identification example, a generative model would start creating new cat pictures instead of deciding if there is a cat in an existing one. This is “Generative AI”.

The beauty of training for LLM's task is that "labeled data" is everywhere around us. Pick a random web page, or even this very blog post. It alone can yield hundreds, if not thousands, of training examples. For example, if you see the string "Pick a random web ", the next word that follows could be "page". It is an equivalent of (picture of a cat, "cat") from our cat picture identifier from earlier. Who created this training example? Me, the author of this post. Of course, when I was writing that sentence, I did not think about it as a training example at all (and was not paid for this). And this is the beauty I was talking about – publicly available text written by humans contains billions of training examples without the need to annotate them. And because you really are talking about billions, you don't have to worry about meticulously curating them – you just throw everything into the pot with a reasonable assumption that all possible sequences and sources are represented in your dataset, because your dataset is literally "everything".

This gets us to the first "L" in "LLM". It stands for "large". These models are large in terms of the amount of data they are trained on and in terms of the number of their parameters, or the weighted variables that the model considers when creating an output. Without going into the math (and with the risk of this analogy having potential holes), a useful way to think about the "parameters" of a model is that they're like neurons in a living creature's brain – the more neurons, the more complex processing and memory the creature is capable of.

You must have noticed that this description of what LLMs do as text completion feels different from the experience you got used to when interacting with ChatGPT, where you ask it questions and the model converses with you. This is true.

Teaching a model to participate in a conversation and follow instructions takes additional fine-tuning and training that we will not cover in this post. Still, holding a conversation is simply generating responses to what the other party is saying and therefore is a type of text completion. Since we mentioned ChatGPT, let's also briefly say what it is. ChatGPT is an LLM that OpenAI optimized for handling conversations with humans and made available to the public.

Back to LLMs. Great – so spending a few millions of $ can get you a model that is trained to predict (or generate) the next word. What is it good for?

First of all, it is entertaining. If you ask it to write a joke about pirates in the style of William Shakespeare or a weather forecast that Ernest Hemingway would write for a local Minnesota TV station on a winter day, the model will generate text that sounds plausible and captures the style you requested (Shakespeare, Hemingway, Minnesota, etc.).

This sounds like an awesome parlor trick. Why is it useful?

LLMs' ability to capture style enables it to "understand" (double quotes are important here, because applying human abilities to the behavior of an inanimate piece of software runs into limitations fast) what a clear, concise, and grammatically correct text looks like and to play an editor to a human writer. This already has some utility in the world of customer support where agents are tasked to communicate with customers all day, not always in a language that they are fluent in, and with the subtleties of cultures that they do not always live in.

But this is only the beginning.

LLMs Can Perform Tasks They Were Not Trained For

LLMs are able to perform tasks that they were not explicitly trained to do and some of these tasks are extremely useful.

Before we proceed, we can't talk about LLMs without mentioning "prompts". The text that you ask the model to complete is called a "prompt". By crafting the right prompt, one can get a model to complete it in a way that achieves a particular task.

Here is one example from the paper Emergent Abilities of Large Language Models published by a team of researchers from Google, DeepMind and Stanford in 2022:

> **The authors entered this prompt to an LLM:**
> **"Review: This movie sucks.**
> **Sentiment: negative**
> **Review: I love this movie.**
> **Sentiment: ".**
> **The model responded with the word "positive".**

This is what a sentiment classifier would say. This LLM demonstrated the ability to predict the sentiment of text without being explicitly trained to do so. Previously, this would take assembling a large set of user reviews, messages, and other examples of text, labeling them with sentiment and training a special purpose ML model that could do only sentiment classification.

How does it do this? The answer is a bit surprising. The same paper says: "there are currently few compelling explanations for why such abilities emerge in the way they do"

Having absorbed great amounts of text created by people somehow allows LLMs to produce text that not only sounds like it makes sense, but represents some real knowledge.

Prompt Engineering

Until we know the exact mechanisms behind such emerging abilities, prompt engineering (the practice of creating and testing optimal prompts for various tasks) feels a lot like SEO – you are poking a system whose rules you don't know in order to get something useful out of it. The difference is that unlike Google, even model creators don't know exactly how an LLM will respond. Practitioners exchange tricks that sometimes sound like magic incantations:

Now that we went over the basic concepts, let's talk about the applications of LLMs, especially in Customer Support.

LLM Applications in Customer Support

LLMs change the practice of Language AI from many "specialist" models that are trained to perform a specific task to a few "generalist" models that have very broad understanding of regularities of human language and can perform multiple tasks, if they are given a reasonable specification of what to do.

These LLMs are expensive to create and run. This means that there are few players that offer them. This also means that using them has non-trivial costs. The number of possible applications that solve problems in various domains is high. The companies that build LLMs (Open AI, Cohere, Anthropic, Google) usually don't build these applications. Instead, they offer access to their LLMs through an API and charge for it. This allows application developers to leverage LLM capabilities without building their own.

Young startups and more mature companies that market "ChatGPT for X" are using Open AI's (or one of the other providers') LLMs via API in order to solve a particular problem.

However, choosing the right problem to solve with LLM is the key.

Remember, LLMs are trained to know what text "makes sense". "Makes sense" does not always mean correct. When an LLM "fibs" an answer, we politely call it "hallucinating". When I was researching IQ testing for my kid, ChatGPT boldly invented a scientific paper's title and author names and used it to back its answers. Only when I insisted it "reasons step by step and makes sure that answer is correct" (this is one of the magical incantations), it backed down and apologized.

If this happened in a customer support setting, it would be problematic.

An extremely simple (yet profound) quote from Richard Socher (an NLP luminary and ex-chief scientist of Salesforce) about choosing use cases for generative/LLM applications: “it’s essential to think about cases when it would take a long time to create some kind of human work artifact, but it would be very quick to verify that it’s correct and useful”.

Building a chatbot that will interact with your customers and give them answers without supervision? Risky! Giving your agents a tool to edit their answers? Promising!

LLMs can be successfully used to improve and scale many operational tasks in customer support. Judging sentiment in customer messages; summarizing a conversation; extracting contact drivers from customer messages; playing an editor to agents that helps them be concise, empathetic, use good grammar and spelling; and creating a draft response for agents to send based on customer’s message are only a few examples.

We will cover some of the applications and the questions that should be considered when buying or building them in subsequent posts.

Now that you understand the fundamental concepts and principles, what use cases do you have for LLMs in your company?

Credits/references

- What Is ChatGPT Doing … and Why Does It Work?, by Stephen Wolfram
- Tenor animated GIF