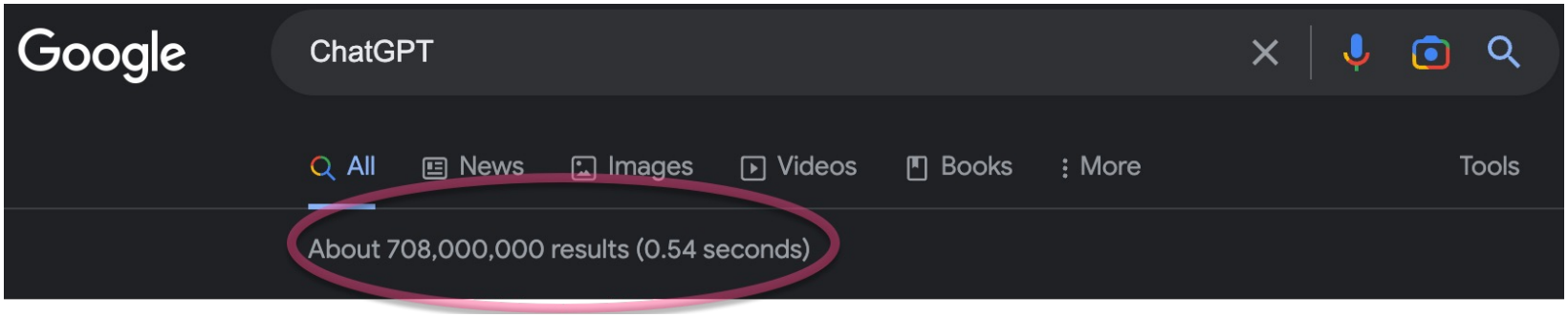
“ChatGPT”: The History and Prospects of Large Language Models

…as relevant to the future of writing instruction…

Suddenly ChatGPT is everywhere:



ChatGPT is a “large language model” (LLM).

But what is a “language model”, large or otherwise?

And where did those LLMs come from,
and where are they going next?

This is a story in two parallel strands.

On both sides, the moral is:

Quantity turns into quality.

These are well-known stories, at least in some circles,
but neither of them has a well-established title -- so we'll go with mine.

I'll call one strand ***AI as applications of the noisy channel model.***

…and I'll call the other ***The surprising alchemy of deep nets.***

Story #1 starts with Claude Shannon in the 1940s, and the “noisy channel model”:

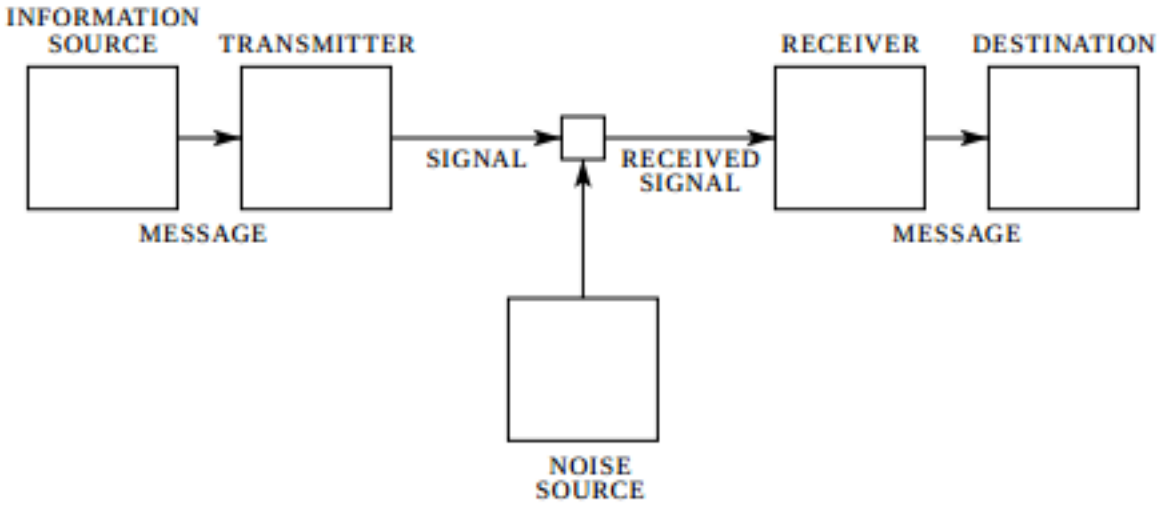


Fig. 1—Schematic diagram of a general communication system.

“A Mathematical Theory of Communication” (1948) introduced ***Information Theory***, and showed how to use encoding and decoding tricks to fix errors due to noise in the transmission channel.

The basic decoding trick is a model of the relative likelihood of possible messages, so that the receiver can pick the most likely message, given the signal.

For Shannon and his colleagues, this was about telecommunications.

But soon, folks realized that the “models of possible messages” trick could be applied to many other problems, including

- Spelling correction
- Optical character recognition
- Speech recognition
- Machine translation
- Image recognition
- etc., etc., etc.

OK, spelling correction and OCR are pretty much like telegraph reception,

but speech recognition?

…and machine translation???

From Jelinek, Bahl, and Mercer 1975:

Most current attempts at automatic speech recognition are formulated in an artificial intelligence framework. In this paper we approach the problem from an information-theoretic point of view. We describe the overall structure of a linguistic statistical decoder (LSD) for the recognition of continuous speech. The input to the decoder is a string of phonetic symbols estimated by an acoustic processor (AP). For each phonetic string, the decoder finds the most likely input sentence. The decoder consists of four major subparts: 1) a statistical model of the language being recognized; 2) a phonemic dictionary and statistical phonological rules characterizing the speaker; 3) a phonetic matching algorithm that computes the similarity between phonetic strings, using the performance characteristics of the AP; 4) a word level search control.

OK, also kind of like correcting characters in a telegram.

(And note that in 1975, “Artificial Intelligence” meant applied logic, not information-theoretically informed applied statistics…)

A few years later, the same group used the same method in machine translation.

The joking explanation:

There's a text in English, but channel noise corrupts the message into French.

So we use a Shannon-style model to undo the corruption,
and restore the message to its intended English-language form.

WEIRD.

But it worked.

And it worked so well that during the 1980s and 1990s,
this is how everyone approached pattern recognition and pattern classification,
using more and more complex models, and more and more training data.

And so "artificial intelligence" came to mean "applied statistics".

And soon, people were using these models as generators,
though mostly as a joke.

For example, a simple statistical language model from 32 years ago,
trained on *Alice in Wonderland*, *Through the Looking Glass*,
and the books of Genesis and Revelation, produced stuff like this:

Alice was beginning to write out a history of all flesh, as God hath judged me, and I will tell you my adventures--beginning from this my oath, when thou fleddest from the engine, and everybody jumped up in alarm, For the Baker had met with again!' "It is this, it turned a back-somersault in at once without waiting for turns, quarrelling all the jurymen on to the top of his mouth went a sharp twoedged sword: and his friends shared their never-ending meal, and the kings of the Jubjub! Keep count, I entreat; You will find I have always been told That the Captain they trusted so well Had only one way up as the Lory hastily. `I thought it must be wasted to-day! "For England expects--I forbear to proceed: 'Tis a maxim tremendous, but trite: And you'd best be unpacking the things which were written in this way when she bare him. And his tail drew the back of it in Dutch-- I said it as she lifted it off, and brought them near unto him. And he was circumcised in the land of Egypt, And all their simple joys, remembering her own child-life, and the seven seals thereof.

OK, time for story #2: **The Surprising Alchemy of Deep Nets**.

This also starts in the 1940s, with a paper called
"A Logical Calculus of the ideas immanent in nervous activity",
which presented a simple model of (pseudo-)neural networks
in which all the neurons were either on or off,
and showed how to translate between such networks
and expressions in propositional logic.

The next step was the Perceptron, in 1958,
which allowed the "neurons" in the network to take on gradient values,
and showed that a simple iterative training algorithm on such nets
could learn a widely-used statistical model called a "linear classifier".

This totally freaked the doyens of Classical Artificial Intelligence,
like Marvin Minsky,
who (co-)wrote a whole book to show that

1. There are simple things that "linear" perceptrons can't do;
2. And non-linearities create dangerous and intractable problems.

This counter-attack was polemically successful for a while,
but it was soon shown that point #2 was wrong –
There are iterative training algorithms, for multi-layered non-linear nets, that

1. generally converge to a solution; and
2. often generalize to data outside their training set.

So there was a “neural net” buzz for a while in the 1980s.

But there were two problems:

1. Not enough data; and
2. not enough computer power.

So most engineers turned back to statistical AI.

For a while.

But Moore's law kept marching on – computers got bigger and faster and cheaper.

Researchers invented new (pseudo-)neural net architectures
and new training methods for them,
able to model data sequences across larger and larger spans of time and space.

The internet boomed, generating lots of data and lucrative applications.

And computer gaming fostered the development of Graphical Processing Units,
good at exactly the massive-scale matrix multiplications that "deep nets" need.

So by 2010 or so, very large versions of these "deep net" models,
trained on very large datasets,
were the best way to create noisy-channel models of "possible messages".

The training-set sizes, the network sizes, and the network complexities,
have continued to grow.

And they've now become good enough that when they're run as generators,
the result is more like Wikipedia than like surrealist poetry.

So what is the "alchemy of deep nets", and why is it "surprising"?

The alchemists were good practical chemists,
but they had no real theories,
just lots of techniques that had been shown to work
(as well as many that didn't work).

And why is this alchemy "surprising"?

Modern deep nets have hundred of billions of parameters,
many orders of magnitude more than the number of training examples.

If these were statistical models,

- they should not converge,
- and if forced to converge, they should not generalize.

But deep nets (sometimes) do,
when the alchemically appropriate methods are used.

Conclusions and predictions?

ChatGPT is not the beginning, and it won't be the end --

Generative LLM rivals and successors are proliferating,
and they will continue to get bigger/better/different.

Among other things, that means that reliable detection is unlikely.

So far, all systems are still imitations of patterns in their training material –
what some have called "high-tech plagiarism",
though that is more than a bit unfair.

In particular, these systems don't "understand" things,
don't form theories, and don't apply logic --
even though they can perform pretty good imitations.

None of that will change along the current growth paths –
maybe in the next generation?