Encode the thought - LLM training for understanding meaning instead of predicting tokens

“Thoughts die the moment they are embodied by words.” A. Schopenhauer

Lyubimkov Dmitri

E-mail: loftlong@gmail.com

2025

Abstract

Current large language models (LLMs) excel at predicting tokens and generating coherent text but often fall short in understanding the deeper meaning behind the words. This paper critiques the token-centric training paradigm of LLMs, arguing that it prioritizes syntax over semantics, leading to models that mimic language without truly comprehending it. Drawing a distinction between words and thoughts, we propose that thoughts are best captured at the level of sentences or larger text fragments, rather than individual tokens. To address this limitation, we explore two alternative training approaches: (1) **Whole Sentence Prediction**, where models predict entire sentences or paragraphs to encourage a focus on meaning, and (2) **Bottleneck Encoder**, where an encoder-decoder architecture compresses text into semantic embeddings, forcing the model to encode the underlying thought. While these approaches present computational and scalability challenges, they offer a promising direction for developing LLMs that move beyond statistical parroting to achieve true understanding. This work invites a reimagining of LLM training paradigms, emphasizing the encoding of meaning over the prediction of tokens.

Table of contents

1. Word and thought

There is no doubt that speech, writing and words are connected with thoughts, they are their consequence. But the opposite is not true - animals do not have speech, but they think their thoughts quite well. Small children do not yet know how to speak, but they have consciousness. We do not believe that our distant ancestors received intelligence at the moment of the emergence of speech. Rather, the mutations of the larynx and vocal cords, which opened the way to articulate sounds, are a consequence of the need to express their thoughts. And finally, there are scientific studies that show the absence of excitation of speech centers during thought processes. We use speech to convey our thoughts to others. Including ourselves, to free the brain from already used thoughts, but to preserve their result. We do not know exactly how the process of thinking occurs in the human brain, what a thought is, but we can separate it from speech.

The human vocal cords, larynx and mouth can produce a limited number of sounds. So we combine these sounds together in various ways and create words from them, each of which stands for a certain entity, action, characteristic, etc. By combining words together we create sentences, which or a set of sentences can already describe our thoughts, so that another can reconstruct them in his brain, regardless of how exactly they look there. What is the carrier of thought? A sound? No. A word? No. A sentence? Already possible. Several consecutive sentences connected to each other? Obviously yes.

Written speech corresponds to spoken speech: letter to sound, word to word, sentences to sentences. Thoughts in written form are conveyed by sentences, paragraphs, sections and chapters.

2. Token and embedding

When we train LLM, we change words to tokens. And at the pre-training stage, we make the neural network predict the next token in texts from a huge corpus of texts. This is completely different from how people exchange thoughts. Because, firstly, the carrier of thoughts is the sentences as a whole, not individual words in them and not the position of these words, which in some languages can be arbitrary. And secondly, people learn not only from texts, they think even before they learn to read, and they impose the texts they hear and read on the genetically determined structure of the brain and the already acquired skills of thinking, consciousness and understanding. LLM are limited by their own architecture, it sees only the text and what they ultimately understand is a consequence of training. Do we train them to predict the next word? Great, most likely they will perfectly understand how this or that language is structured, how words are related to each other. They will become good translators. But they will be able to understand the thought contained in the texts with great limitations, because the thoughts are not in words, but in sentences. Maybe this is the main problem of modern LLMs, that they learn into embeddings not thoughts, but syntax, not ideas, but words. And as a result, instead of superintelligence we get a statistical parrot. Let's consider two possible ways of teaching LLMs the thoughts hidden behind words.

3. LLM with whole sentence prediction

Let's leave everything that LLM has now - architecture, infrastructure, text corpus, but at the pre-training stage we will rework everything so that the model predicts not the next word, but the thought contained in the current text fragment. What do we consider this thought? At least all tokens up to the end of the current sentence. Possibly all tokens up to the end of the paragraph. If we use the method from the article https://github.com/loftyara/LLMagogy for training LLM, then thoughts can be marked manually (or using another neural network).

The loss function in this case should be calculated as the sum of the loss from all words/tokens, and each word/token should have its own additional distance coefficient. The further this word is from the current position in the text, the smaller this coefficient. We still try to predict mainly the nearest tokens, which allows us to hope that LLM will be no less accurate than before and at the same time forces it to look ahead and understand not only the syntax of the language, but also the thoughts embedded in the text.

4. LLM with bottleneck encoder

In this variant, the use of the encoder/decoder combination is mandatory. Variants with only an encoder or only a decoder are excluded. In this case, the encoder is not intended for individual tokens. The sentence (paragraph, etc.) is transformed into embeddings as a whole. That is, the encoder transforms the sequence of tokens into a certain coordinate in the space of meanings. Which is synonymous with understanding the thought embedded in this fragment as a whole. And the decoder, in turn, must restore the original sequence of tokens.

Options for how this can be done are described in the article https://github.com/loftyara/CLLM_void_zones

5. PS.

On February 18, 2025, the article “Cramming 1568 Tokens into a Single Vector and Back Again: Exploring the Limits of Embedding Space Capacity” was published. The authors managed to reduce the size of the token embedding sequence by 1500 times with the ability to restore the initial values. Moreover, they did not even use complex neural network models for this. This text contains 1000 - 1500 words and approximately the same number of tokens. That is, the entire document can be fit into single “thought”, which is even excessive for our purposes.