Università degli Studi di Torino

Corso di Laurea Magistrale in Informatica

Concrete Numeric Representations in LLM Embeddings

Tesi di Laurea

Relatore/Relatrice
Prof. Di Caro Luigi

Correlatore/Correlatrice
Dr. Torrielli Federico

Candidato/a
Gentiletti Emanuele
900831

Anno Accademico 2024/2025

Contents

Introduction **1**

The Transformer architecture and vector representations **3**
The inductive bias of Tokenization 3
Reification as computed embeddings - xVal 4

Paragraphs yet to contextualize - not a real section **7**

Bibliography **9**

Introduction

This work started with a simple premise: why are LLMs bad at math?

This is not really a hard question to answer. Most of the LLMs to date are not built with that purpose in mind, and can rely on tool calling to give good answers to quantitative and numerical questions.

There is a tremendous investment in computing resources that is directed towards arithmetic operations that make up the inner workings of LLMs, computations that the LLMs themselves aren't capable of leveraging to answer arithmetic questions. It feels like witnessing a fundamental disconnection, where the LLM is segregated from the capabilities that make its own functioning possible.

Savant syndrome is a very rare disorder. It manifests primarily in people with autism spectrum disorders (Murray, 2010) or after traumatic episodes. The people affected by it possess extraordinary qualities in certain areas, like arts, music or mathematics, while usually showing significant impairment in others. One of the possible areas in which savants may show exceptional aptitude is calculation: calendrical savants are able to instantly know the day of the week of dates far in the future. These skills are unlikely to be the product of algorithmic calculation (Cowan & Frith, 2009), so alternative hypotheses emerged.

What I propose here is that the Savant condition can be seen as a parallel to the bridging of this capabilities gap in LLMs. In particular, what is taken in consideration here is the use of concrete representations as described in (Murray, 2010), where abstract numerical concepts are transformed into "highly accessible concrete representations" that can be directly manipulated rather than computed through algorithmic steps. This reification process - the conversion of abstract concepts into concrete entities - appears to provide savants with immediate access to numerical relationships that would otherwise require complex calculations.

This is not meant necessarily to give a comprehensive explanation of the phenomenon on an empirical basis, as that would be hard to establish from the basis of current knowledge about both savant cognition and neural network representations. Rather, it serves as a conceptual framework for exploring whether similar representational advantages can be induced in artificial systems.

This idea is explored in two ways:

- by a literature review, that is meant to clarify what can function as concrete representations in this context
- by an exploration of numerical embeddings, that is meant to show whether the learned representation of current language models already tends to conform to certain geometrical objects or structures. We show that there is remarkable structure and patterns in the learned representation of current LLMs.

The Transformer architecture and vector representations

The inductive bias of Tokenization

Modern LLMs are built on the Transformer architecture (Vaswani et al., 2023), which operates by converting input text into sequences of discrete tokens that are then mapped to high-dimensional vector representations. This initial tokenization step creates an inductive bias that shapes how the model processes information (Ali et al., 2024) (Singh & Strouse, 2024), with significant implications for the application of the numerical data to arithmetical tasks.

The most used algorithm for tokenization is currently Byte-Pair Encoding, which, given a fixed vocabulary size, starts with individual characters and iteratively merges the most frequently occurring pairs of adjacent tokens until the vocabulary limit is reached. This process naturally creates longer tokens for common substrings that appear frequently in the training data. For numbers, this means that frequently occurring numerical patterns like "100", "2020", or "999" might become single tokens, while less common numbers get broken into smaller pieces. The result is an idiosyncratic and unpredictable tokenization scheme where similar numbers can be tokenized completely differently based purely on their frequency in the training corpus. While GPT-2 used to have a purely BPE tokenizer, the successive iteration of GPT and generally more recent models either tokenize digits separately (so as $'1234' \rightarrow [1, 2, 3, 4]$), or tokenize clusters of 3 digits, encompassing the integers in the range 0-999.

Most of the tokenizers right now do L2R (left-to-right) clustering, meaning that a number such as 12345 would be divided in two tokens, 123 and 45. It has been shown (Singh & Strouse, 2024) that this kind of clustering leads to a lesser arithmetic performance, as the grouping doesn't match the positional system's <way of calculating?>. An even more surprising development is that forcing the R2L token clustering of numbers in models already trained with L2R clustering through the use of commas in the input (ex. $12, 345$) leads to big improvements in arithmetic performance (Millidge, n.d.-b). Despite the model learning representations adapted to work with a L2R token clustering strategy, forcing a R2L clustering at inference time shows substantial improvements in arithmetic tasks, which means that despite being learned through an unfavorable tokenization approach, the numeric representations retain the properties that allow for the performance to improve when the clustering scheme is corrected.

There can be different hypotheses on why this might be, for example:

- Arithmetic operations would still work locally in the 0-999 range, which allows for a correct reading on them and possible generalization on a larger scale.
- The forced tokenization also happens in the data, as numbers are often separated by punctuation in clusters of 3 digits, right to left, for legibility reasons (Singh & Strouse, 2024)

Still, we are left with the fact that the learned representations work better for a tokenization strategy different from the one the model was trained for. At the very least, the data being biased towards a R2L representation (in the form of using the Arabic number system and adopting legibility rules that accommodate right to left calculations) lead to embeddings that maintain that bias even when learned in a L2R fashion.

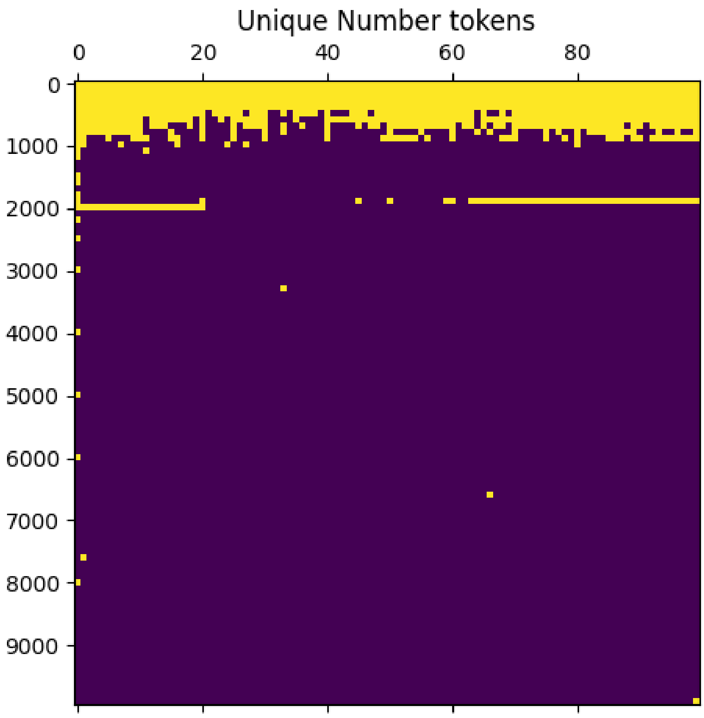

Figure 1: GPT-2 number tokenization. Each row represents 100 numbers, yellow squares mean that the number is represented by a single token, purple ones by multiple (Millidge, n.d.-a)

Table 1: Language models with their respective tokenization strategy for numbers.

Model	Strategy
LLaMA 1 & 2	single digit
LLaMA 3	L2R chunks of 3 digits
OLMo 2	L2R chunks of 3 digits
GPT-2	pure BPE
GPT-3.5/4	L2R chunks of 3 digits
Claude 3/4	R2L chunks of 3 digits

Reification as computed embeddings - xVal

There have been other, more comprehensive approaches to the improvement of the representation of numeric values. xVal is a notable one, as its approach encompasses real numbers beyond just integers and does away with learning different representation for each number.

The idea is maximizing the inductive bias in the representation by having embeddings that are computed based on the number to be represented. Numerical values represented by a single embedding vector associated with the [NUM] special token.

This fits very well with the idea of reification: the embedding is no longer just a representation, but it contains and has properties of the object it represents.

The model uses two separate heads for number and token predictions. If the token head predicts a [NUM] token as the successor, the number head gets activated and outputs a scalar. The rest of the weights in the transformer blocks are shared, allowing the learning of representations that are useful for both discrete text prediction and continuous numerical prediction. This means the model develops number-aware internal representations throughout all its layers, not just at the output. The shared weights force the model to learn features that work for both linguistic and mathematical reasoning simultaneously.

The approach is shown to improve performance over a series of other techniques, mostly using a standard notation to represent numbers.

> Nota: qua l'incastro dovrebbe essere con (Murray, 2010) e il caso di DT, descritto sotto. Il problema è che il "panorama numerico" descritto sembra essere più discreto che continuo, mentre qua si parla della codifica di numeri continui. Potrebbe essere preferibile incastrarsi direttamente alla parte di analisi, dove vengono visualizzati gli embedding nel range 0-9999, anche se xVal è un buon esempio di reificazione.

A case study of a Savant patient, DT (Murray, 2010), has been reported of having a mathematical landscape with the following characteristics:

- Has sequence-space synesthesia with a "mathematical landscape" containing numbers 0-9999
  - sequence-space synesthesia: *spatial sequence synesthesia consists of visualising certain sequences in physical space.*
- Each number has specific colors, textures, sizes, and sometimes movements or sounds
- Prime numbers have special object properties that distinguish them from other numbers
- Arithmetic calculations happen automatically - solutions appear as part of his visual landscape without conscious effort
- fMRI studies showed that even unstructured number sequences had visual structure for DT

Paragraphs yet to contextualize - not a real section

In (Mottron et al., 2006), the hypothesis is also that the capabilities of the savant might come from privileged access to lower-level perceptual processing systems that have been functionally re-dedicated to symbolic material processing. This suggests that mathematical savants may bypass high-level algorithmic reasoning entirely, instead leveraging perceptual mechanisms that can directly recognize patterns in numerical relationships - much like how we might instantly recognize a face without consciously processing its individual features. There are also arguably similar mechanisms already implemented in LLMs, although usually employed in the context of <?> gradient normalization, in the form of skip connections.

Bibliography

Ali, M., Fromm, M., Thellmann, K., Rutmann, R., Lübbering, M., Leveling, J., Klug, K., Ebert, J., Doll, N., Buschhoff, J. S., Jain, C., Weber, A. A., Jurkschat, L., Abdelwahab, H., John, C., Suarez, P. O., Ostendorff, M., Weinbach, S., Sifa, R., … Flores-Herr, N. (2024, March 17). *Tokenizer Choice For LLM Training: Negligible or Crucial?* https://doi.org/10.48550/arXiv.2310.08754

Cowan, R., & Frith, C. (2009). Do calendrical savants use calculation to answer date questions? A functional magnetic resonance imaging study. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1522), 1417–1424. https://doi.org/10.1098/rstb.2008.0323

Millidge, B. (n.d.-a). *Integer tokenization is insane*. Retrieved June 26, 2025, from http://www.beren.io/2023-02-04-Integer-tokenization-is-insane/

Millidge, B. (n.d.-b). *Right to Left (R2L) Integer Tokenization*. Retrieved June 26, 2025, from http://www.beren.io/2024-07-07-Right-to-Left-Integer-Tokenization/

Mottron, L., Lemmens, K., Gagnon, L., & Seron, X. (2006). Non-algorithmic access to calendar information in a calendar calculator with autism. *Journal of Autism and Developmental Disorders*, *36*(2), 239–247. https://doi.org/10.1007/s10803-005-0059-9

Murray, A. L. (2010). Can the existence of highly accessible concrete representations explain savant skills? Some insights from synaesthesia. *Medical Hypotheses*, *74*(6), 1006–1012. https://doi.org/10.1016/j.mehy.2010.01.014

Singh, A. K., & Strouse, D. J. (2024, February 22). *Tokenization counts: The impact of tokenization on arithmetic in frontier LLMs*. https://doi.org/10.48550/arXiv.2402.14903

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, August 1). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762