Universidad ORT Uruguay
Facultad de Ingeniería

Exploring Attention Patterns and Neural Activations in Transformer Architectures for Sequence Classification in Context Free Grammars

Entregado como requisito para la obtención del título de Ingeniería en Sistemas

Matías Molinolo De Ferrari - 231323

Tutores: Dr. Sergio Yovine, Dr. Franz Mayr

2024

Declaración de Autoría

Yo, Matias Molinolo De Ferrari, declaro que el trabajo que se presenta en esta obra es de mi propia mano. Puedo asegurar que:

- La obra fue producida en su totalidad mientras realizaba el Proyecto Final de Ingeniería en Sistemas;
- Cuando he consultado el trabajo publicado por otros, lo he atribuido con claridad;
- Cuando he citado obras de otros, he indicado las fuentes. Con excepción de estas citas, la obra es enteramente mía;
- En la obra, he acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, he explicado claramente qué fue contribuido por otros, y qué fue contribuido por mi;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Matias Molinolo De Ferrari

dd-10-2024

Agradecimientos

{DEDICATORIA}

{AGRADECIMIENTOS}

Abstract

Cuerpo del Abstract.

Abstract Español

Cuerpo del Abstract.

Palabras clave

tag1; tag2; tag3

Key words

tag1; tag2; tag3

Contents

1 Introduction

Large Language Models (LLMs) have been a topic of interest in the field of Computer Science for the past few years, and more recently, with the release of ChatGPT [1] by OpenAI, they have become a topic of interest for the general public too.

These models are based on an architecture called Transformer [2], a type of Artificial Neural Network (ANN) well suited to process sequences, such as text. These models have grown exponentially, as seen in 1.1, both in size and complexity, in the last years, and have shown to achieve state-of-the-art results in a wide variety of Natural Language Processing (NLP) and Natural Language Understanding (NLU) tasks.

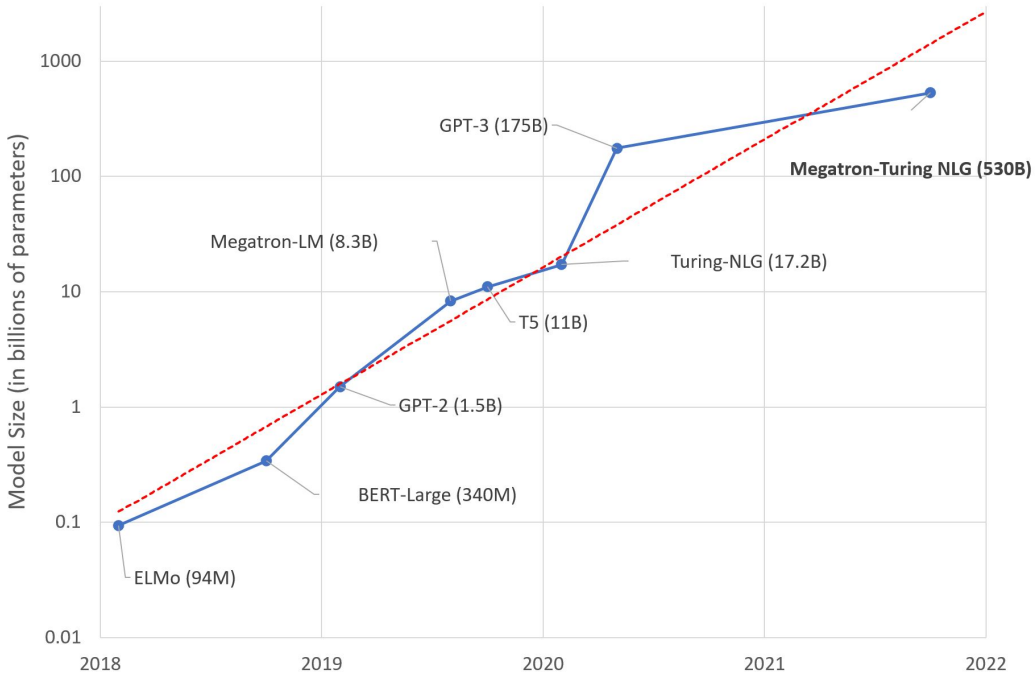

Figure 1.1: Model Sizes (2018–2021) [3]

However, despite their state-of-the-art performance, the inner workings of these models are not yet fully understood and these models are still considered opaque or *black-box* [4], which is a problem for their adoption in critical applications, such as healthcare or finance where decisions need to be explainable and interpretable.

Moreover, there is still a boundary to the capabilities of these models, regarding which problems can or cannot be solved by them and what can be learned and expressed by these models.

Strobl et. al [5] defines two lines of work done regarding this problem: *expressivity* and *trainability*. This work will focus on the former, as we aim to determine, using a *white-box* approach, by looking at the attention mechanism and neural activations, whether Transformers are able to learn and classify sequences belonging to a formal language, more specifically, context-free grammars (CFGs) such as Dyck languages.

Chomsky's hierarchy [6] classifies CFGs as those languages that can be represented by a nondeterministic pushdown automaton, a class of automata that is more expressive than finite-state machines but less so than Turing machines [7]. Pérez, Barceló and Marinkovic propose that architectures based on self-attention, such as Transformers, are Turing complete [8], therefore, this leads us to believe that CFGs can be expressed by these neural language models.

Based on the aforementioned, we seek to discuss whether attention patterns and neural activations in these architectures can be used to explain the model's classifications, and if these explanations can be leveraged to improve the model's performance and interpretability.

Outline

Chapter 2 will introduce the concepts behind formal languages and context-free grammars, focusing especially on the Chomsky Hierarchy and Dyck-k languages. Chapter 3 will introduce the Transformer architecture, with a focus towards the attention mechanism. Chapter 4 will discuss related works and the state-of-the-art in the field of explainable AI and formal languages Chapter 5 focuses on the experimental setup, the dataset used, the model architecture, the training process and the obtained results. Chapter 6 will discuss the results obtained and the implications of these results. Chapter 7 will summarize the work done and propose future work.

2 On Formal Languages

Hopcroft, Motwani and Ullman define a language $\mathcal{L}$ as a set of strings chosen from Σ^*, where Σ is a particular alphabet. Σ^* denotes the *universal language*, which is the language formed by all possible sequences over an alphabet Σ. An alphabet is defined as a finite, nonempty set of symbols. [9]

Furthermore, they define a string or word as a finite sequence of symbols chosen from an alphabet Σ. Therefore, a word can also be seen as a concatenation of symbols that start with the *empty* or *identity* symbol, ϵ.

However, not every word formed by the alphabet necessarily belongs to a language, since for each language, a specific set of rules exist that define *membership* - whether a word belongs to the language or not.

Furthermore, it is important to note the existence of the *empty language*, $\varnothing$, that is, the language that does not contain any words, not even the empty one - $\epsilon \notin \varnothing$.

The only constraint on what can be a language is that, by definition, alphabets are finite, therefore, even though there might be an infinite amount of strings in a language, they all stem from a finite alphabet [9].

Chomsky defines a grammar as a "device that enumerates the sentences of a language" and proposes a set of restrictions that limit these grammars to different types of automata, in a way that the "languages that can be generated by grammars meeting a given restriction constitute a proper subset of those that can be generated by grammars meeting the preceding restriction" [6].

Furthermore, Chomsky proposes that these grammars, ordered by increasing restrictions, can be recognized by different automata - Turing machines, linear-bounded automata (LBA), pushdown automata (PDA) or Finite State Machines (FSM), as seen in 2.1.

For this work, we will focus on Type-2, or Context-Free Grammars, which can be recognized by pushdown automata, which have a "natural, recursive notation" [9]. For example, palindromes are a context-free language, as the grammar that generates it is also context-free.

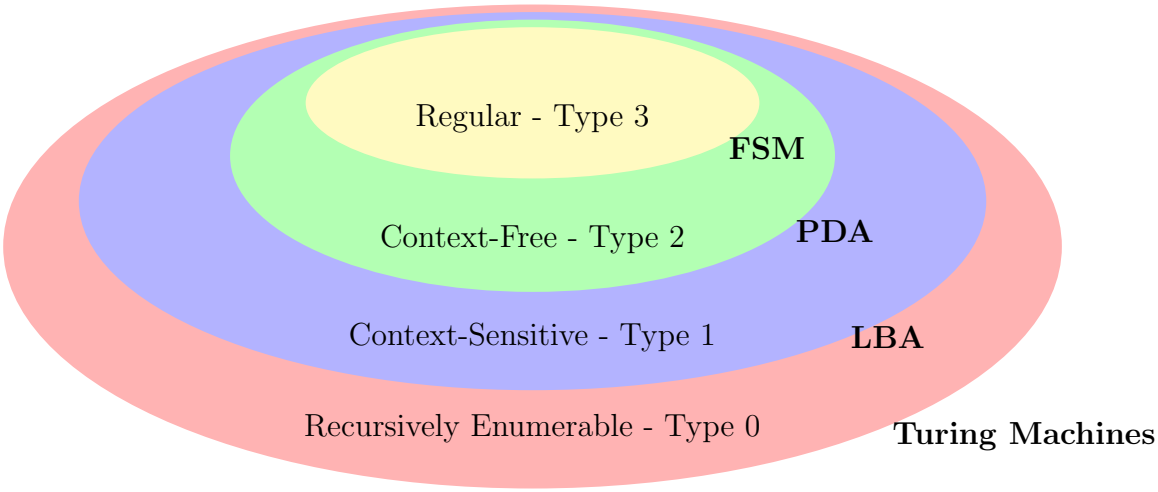

Figure 2.1: Chomsky Hierarchy

2.1 Context-Free Grammars

A context free grammar $\mathcal{G}$ has four main components [9][10]:

1. A finite set of *variables* $\mathbf{V}$ - these variables represent a language (a set of strings). We note $\mathbf{V}^*$ as the Kleene closure of the set of variables - in essence, the concatenation of zero or more repetitions of elements in the set.

2. A finite set of symbols $\mathbf{T}$, called *terminals*. Note that $\mathbf{T} \subset \mathbf{V}$. Once again, $\mathbf{T}^*$ is the Kleene closure of the set.

3. A *start symbol* $\mathbf{S} \in \mathbf{V} - \mathbf{T}$.

4. A finite set of *productions*, $\mathbf{P} \subset (\mathbf{V} - \mathbf{T}) \times \mathbf{V}^*$. These represent the recursive definition of the language. Each production consists of the following:

   (a) A variable, called the *head* that is partially defined by the production.

   (b) A production symbol $\rightarrow$

   (c) A string of zero or more terminals and variables, called the *body*.

Therefore, a grammar $\mathcal{G}$ can be expressed as a four-tuple $(\mathbf{V}, \mathbf{T}, \mathbf{P}, \mathbf{S})$.

Given 2 words, $u, v \in \mathbf{V}^*$, we say that $u \rightarrow v$ if there exists a derivation, or sequence of words in $\mathbf{V}^*$ such that $u_{i-1} \rightarrow u_i$ for $i = 1, \ldots, k$ and $u_0 = u$ and $v = u_k$. The existence of a derivation is noted by $u \xrightarrow{*} v$

Furthermore, we define the *language* $\mathcal{L}$ *generated by* $\mathcal{G}$ as the following set:

$$\mathcal{L}(\mathcal{G}) = \{w \in \mathbf{T}^* | \mathbf{S} \xrightarrow{*} w\}$$

Should X be a variable in $\mathcal{G}$, then:

$$\mathcal{L}_{\mathcal{G}}(X) = \{w \in \mathbf{T}^* | X \xrightarrow{*} w\}$$

From this, we derive that $\mathcal{L}(\mathcal{G}) = \mathcal{L}_{\mathcal{G}}(S)$. If two grammars generate the same language, they are considered *equivalent* [10].

2.2 Dyck-k Languages

Dyck-k languages are a canonical family of context-free languages, composed of strings of balanced parentheses. A Dyck-k language is defined as a set of strings over the alphabet $\Sigma = \{a_1, \bar{a_1}, \dots a_n, \bar{a_n}\}$, where each string is a sequence of $2k$ parentheses, such that the parentheses are balanced. That is, for each string $w \in \Sigma^*$, the number of opening parentheses is equal to the number of closing parentheses, and for each prefix w' of w, the number of opening parentheses is greater than or equal to the number of closing parentheses.

The language of Dyck-k is denoted by D_k and is defined by the following grammar:

$$\begin{aligned}
\mathbf{V} &= \{S\} \\
\mathbf{T} &= \{a_1, \dots a_n\} \cup \{\bar{a_1}, \dots, \bar{a_n}\} \\
\mathbf{P} &= \{S \to a_i S \bar{a_i} S | \epsilon\} \text{ for } i = 1, \dots, n \\
\mathbf{S} &= \epsilon
\end{aligned}$$

In this case, $\mathbf{T}$ is called a *matched alphabet*, as each symbol a_i has a corresponding closing symbol $\bar{a_i}$.

We say that Dyck-k languages are canonical because they are the simplest form of context-free languages and can be used as a building block for all other context-free languages. This is known as the Chomsky-Schützenberger Theorem [11] and states that a language $\mathcal{L}$ over an alphabet Σ is context-free iff there exists:

- a matched alphabet $T \cup T'$,
- a regular language R over $T \cup T'$,
- a homomorphism $h : (T \cup T')^* \to \Sigma^*$

such that $\mathcal{L} = h(R \cap D_k)$ for some k.

3 Transformer Architecture

4 Bibliography

[1] OpenAI, Nov 2022. [Online]. Available: https://openai.com/index/chatgpt

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[3] J. Simon, "Large language models: A new moore's law?" Oct 2021. [Online]. Available: https://huggingface.co/blog/large-language-models

[4] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, and X. Carreras, Eds. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 107–117. [Online]. Available: https://aclanthology.org/D16-1011

[5] L. Strobl, W. Merrill, G. Weiss, D. Chiang, and D. Angluin, "What formal languages can transformers express? a survey," 2024.

[6] N. Chomsky, "On certain formal properties of grammars," *Information and Control*, vol. 2, no. 2, pp. 137–167, 1959. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019995859903626

[7] N. Chomsky and M. Schützenberger, "The algebraic theory of context-free languages*," in *Computer Programming and Formal Systems*, ser. Studies in Logic and the Foundations of Mathematics, P. Braffort and D. Hirschberg, Eds. Elsevier, 1963, vol. 35, pp. 118–161. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0049237X08720238

[8] J. Pérez, P. Barceló, and J. Marinkovic, "Attention is turing-complete," *Journal of Machine Learning Research*, vol. 22, no. 75, pp. 1–35, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-302.html

[9] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[10] J. van Leeuwen, Ed., *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*. Elsevier and MIT Press, 1990. [Online]. Available: https://www.sciencedirect.com/book/9780444880741/formal-models-and-semantics

[11] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages: Volume 1. Word, Language, Grammar*, ser. Handbook of Formal Languages. Springer, 1997. [Online]. Available: https://books.google.com.uy/books?id=yQ59ojndUt4C

5 Annexes

5.1 Annex 1

5.2 Annex 2