Lengua y Sociedad. Revista de Lingüística Teórica y Aplicada,
Vol. 23, n.º 2, julio-diciembre 2024, pp. 599-622,
ISSN-L 1729-9721; eISSN: 2413-2659
https://doi.org/10.15381/lengsoc.v23i2.28972

The role of linguistics in artificial intelligence (AI)

El papel de la lingüística en la inteligencia artificial (IA)

O papel da linguística na inteligência artificial (IA)

Celso Vargas-Elizondo
The Costa Rica Institute of Technology, Costa Rica
celvargas@itcr.ac.cr
https://orcid.org/0000-0002-1701-6186

Abstract

In this paper, I analyse the contributions of the Natural Language Processing (NLP) perspective to the development of Artificial Intelligence (AI). It is divided into four main sections. In the first one, I introduce a brief and general conceptualisation of AI, aiming to indicate the place of NLP in the general perspectives of AI. In the second section, I analyse the theoretical linguistic elements that played a relevant role in the evolution of NLP. In the third section, I introduce the relationships between computation and natural language analysis to provide a general concept of their connection. Finally, in the fourth section, I introduce a brief characterisation of Large Language Models (LLMs), Foundational Models (FMs) and their relations with NLP. So, this paper aims to provide a general perspective of this complex field, emphasising the milestone issues.

Key words: artificial intelligence; natural language processing; linguistics; neural networks; foundational models.

Resumen

En este artículo analizo las contribuciones del enfoque del procesamiento de lenguaje natural (PLN) al desarrollo de la Inteligencia Artificial (IA). Se compone de cuatro secciones principales. En la primera se ofrece una breve conceptualización de la inteligencia artificial con el objetivo de ubicar el lugar del procesamiento del lenguaje natural en IA. En la segunda sección se proporcionan algunos de los elementos lingüísticos teóricos más relevantes para entender la importancia del procesamiento del lenguaje natural y su evolución. En la tercera, presentamos algunos elementos teóricos importantes para entender la forma en la funciona el procesamiento del lenguaje natural. Finalmente, en la cuarta sección, presentamos brevemente las principales características de los modelos de lenguaje de gran tamaño (LLMs) y de los modelos fundacionales (FMs) y su relación con el procesamiento del lenguaje natural. El objetivo general del artículo es proporcionar una visión general de este complejo campo de investigación, poniendo énfasis en los aspectos más relevantes.

Palabras clave: inteligencia artificial; procesamiento de lenguaje natural; lingüística; redes neuronales; Modelos Fundacionales.

Resumo

Neste artigo analiso as contribuições da abordagem do processamento de linguagem natural (PNL) para o desenvolvimento da Inteligência Artificial (IA). É composto por quatro seções principais. A primeira oferece uma breve conceituação de inteligência artificial com o objetivo de localizar o lugar do processamento de linguagem natural na IA. A segunda seção fornece alguns dos elementos linguísticos teóricos mais relevantes para compreender a importância do processamento da linguagem natural e sua evolução. Na terceira, apresentamos alguns elementos teóricos importantes para compreender como funciona o processamento de linguagem natural. Por fim, na quarta seção, apresentamos brevemente as principais características dos Large Language Models (LLMs) e dos modelos básicos (FMs) e sua relação com o processamento de linguagem natural. O objetivo geral do artigo é fornecer uma visão geral deste complexo campo de pesquisa, enfatizando os aspectos mais relevantes.

Palavras-chave: inteligência artificial; processamento de Linguagem Natural; lingüística; redes neurais; modelos fundamentais.

Received: 29/03/2024 Accepted: 28/07/2024 Published: 30/12/2024

1. Introduction

The amazing achievements in Artificial Intelligence (AI) in this third decade based on self-supervised neural networks (NNs) changed many of our expectations of the potentialities of AI. In these achievements, Natural Language Processing (NLP), the computational approach to natural language has played an important role. What is interesting is that NLP and the construction of programming languages start from the same source: The "Models for the Description of Language" by Noam Chomsky (1956). This publication inspired the construction of programming languages with well-defined mathematical properties, but at the same time opened the door for a new paradigm of research in linguistics: generative (transformational) grammars and other approaches influenced by the generative paradigm with the same mathematical concerns. The history of these developments is a complex issue. So, I emphasise some of the key linguistics milestones aimed at understanding the role of NLP in developing the new chatbots and virtual assistants. Four main issues are addressed to understand the relationships between NLP and AI. The first issue provides a brief approach to AI. I start from Russell and Norvig (2010) who classified philosophical and scientific positions into four main categories. The second summarises some of the elements of the generative-based linguistic approach. The third issue concerns the computational approach of NLP. Finally, I introduce some general considerations on Large Language Models (LLMs), Foundational Models (FMs) and a very brief comment on the relationship between FMs and NLP. Our discussion is rather informal than formal. I introduce only the symbols needed to understand the issue.

2. Four main research programmes on Artificial Intelligence

In November 2022, OpenAI popularised the virtual assistance ChatGPT-3. Even though Google, Amazon and others have been using Artificial Intelligence (AI) for virtual chatbots and assistants, also called Large Language Models (LLMs), ChatGPT-3 and ChatGPT-4 have become acknowledged as representing an important achievement in AI. As indicated by Naveed *et al.* (2024) more than 50 of these models are available from 2019 to 2024 for different areas, different purposes, and scopes, including art creation, translation, and music composition. A dramatic increase in the number of these applications has been observed since 2021. However, this new generation of virtual chatbots and assistants is only a part of a larger area of AI, many of its applications we use routinely in internet research, email classification, in some alerts, among others. So, AI is one of the most dynamic fields currently.

For a better understanding of the philosophical, technological, and social changes associated with these progresses in AI, it is necessary to provide some conceptual context. With this aim, I find very relevant the categorisation of positions on AI introduced by Russell and Norvig (2010). These authors classified the different philosophical and scientific positions into four different categories according to the goal pursued by them. Adapted from these authors is the following table:

Table 1
Classification of philosophical and scientific positions on AI

Categories	Human-based approaches	Rationally based approaches
Thought based approaches	Thinking humanly	Thinking rationally
Behaviour-based approaches	Acting humanly	Acting rationally

Note. Adapted from Russell and Norvig (2010, p. 2)

As observed, two positions consider AI´s goal limited to human beings, while the other two positions consider AI as consisting of simulating rational and behavioural situations beyond humans, including herd behaviour, for example. Let us consider with more details each one of these positions. The first is *thinking humanly*, that endorses the goal of AI in the following way:

> We need to get inside the actual workings of human minds. There are three ways to do this: through introspection—trying to catch our thoughts as they go by; through psychological experiments—observing a person in action; and through brain imaging—observing the brain in action. Once we have a sufficiently precise theory of the mind, it becomes possible to express the theory as a computer program. If the program's input-output behaviour matches corresponding human behaviour, that is evidence that some of the program's mechanisms could also be operating in humans. (Russell and Norvig, 2010, p. 3).

It is a tremendous scientific research programme that endorses, mainly the inter and multidisciplinary programme called "Neuroscience". Understanding the brain and how behavioural and cognitive processes emerge from the brain (and interact with it), and the nervous systems, face many challenges currently. However, much progress has been achieved in understanding, both, behavioural and cognitive as associated with different elements at the level of neurons and neural networks, such as neurotransmitters, neuromodulators, modulation of cortical responses and other substances involved in the brain processes. Dowling (2018) provides a systematic approach that includes the main results obtained within this important field at that date. In addition to this, one of the most important technological results is the "Brain computer interfaces" (BCIs). In 2013, Javaid (2013) introduced the main developments in BCI in the following way:

> A Brain-computer interface, sometimes called a direct neural interface or a brain-machine interface, is a direct communication pathway between a brain and an external device. It is the ultimate in the development of human-computer interfaces or HCI. BCIs being the recent development in HCI, there are many realms to be explored. After experimentation, three types of BCIs have been developed namely Invasive BCIs, Partially-invasive BCIs, and Non-invasive BCIs. (p. 2)

It is expected that this year (2024) several breakthroughs in neuroscience take place, mainly in the medical sector, including of course, BCIs: "we can expect to see BCIs used to treat a wide range of neurological disorders, including paralysis, epilepsy, and chronic pain" (Foothills Neurology, 2024, n.p.). Despite these achievements, there remains to be done to make it possible to achieve the goal of simulating on computers how humans think.

Thinking rationally, several traditional and modern perspectives form what is called the "symbolic paradigm" in AI. From a traditional perspective, the laws of thought indicate the efforts of philosophers and mathematicians to capture the nature of thinking by logical systems, such as Aristotle's syllogistic, or the complex logical systems constructed by Leibniz. Of course, it includes modern development in symbolic logic, especially the logistic tradition, and several interesting products such as expert systems and decision-making systems. In computer science, it was also the dominant paradigm during the second part of the XX century. However, the emergence of the "sub-symbolic paradigm", especially during the 1990s, has rivalled its position in computer science.

"Connectionism" is the new paradigm. As we will see, it was within the connectionism paradigm that the new revolution in AI emerged. Compared with sub-symbolic approaches, the symbolic approach faces two main problems as indicated by Russell and Norvig:

> There are two main obstacles to this approach. First, it is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain. Second, there is a big difference between solving a problem "in principle" and solving it in practice. [...] Although both of these obstacles apply to any attempt to build computational reasoning systems, they appeared first in the logicist tradition. (Russell and Norvig, 2010, p. 4)

Acting rationally was introduced by Russell and Norvig as the most promising approach to describe and promote the development of AI. It is more pragmatic than the other approaches in the sense that AI can be used to describe different kinds of rationality involved in the simulation and understanding of different behavioural-based processes. But also, it is easier to describe the job of AI in terms of two basic concepts for the design and implementation of AI systems: "intelligent agent" and "rational agent". The first of these concepts emphasises the structure of an agent that interacts with an environment by sensors and actuators, gets information from it, and makes decisions within the scope of its expertise; the second one provides metrics to assess the performance of the intelligent agent, including desirable and optimised actions. Given that rationally, as introduced by Russell and Norvig includes criteria of success, "agent´s prior knowledge of the environment", the kind of actions that can be performed by the system and "the agent's percept sequence to date", they proposed the following definition of a rational agent: "For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has" (Russell and Norvig, 2010, p. 37).

The rational agent is an integral element of the design and implementation of intelligent agents and our current conception of artificial intelligence. Let A and B be two intelligent computer programmes, then we may say that A is more intelligent than B if the way A interacts with an environment and makes decisions is more desirable and optimum than B´s (providing some parameters for "desirable" and "optimum"). In the implementation of intelligent systems, the more desirable goals are targeted, so the intelligent agent should act rationally. So, "(t)he job of AI is to design an agent program that implements the agent function—the mapping from percepts to actions", and a function for evaluating its performance (Russell and Norvig, 2010, p. 46).

However, the revolution in AI that we have observed since 2021 relates to the development of LLMs framed within the fourth category of AI: **acting humanly.** One of the best representatives of this category is the Turing Test. As pointed out by Alan Turing (1950), we can claim that an AI system acts as humans in the case in which it is not possible for a human being to claim that he is interacting with a computer or with a human being. Accordingly, the computer would need to possess the following capabilities:

- natural language processing to enable it to communicate successfully in English;
- knowledge representation to store what it knows or hears;

- automated reasoning to use the stored information to answer questions and to draw new conclusions.
- machine learning to adapt to new circumstances and to detect and extrapolate patterns.
- […] To pass the total Turing Test, the computer will need
- computer vision to perceive objects, and
- robotics to manipulate objects and move about. (Russell and Norvig, 2010, pp. 2-3).

The integration of these six capabilities remains to be achieved, but amazing progress has been made in these areas. As mentioned above, ChatGPT-4 achieved a lot in natural language understanding, knowledge representation, automated reasoning, and machine learning. We will observe more progress soon. For example, achieving complete autonomy in cars would provide feasible models for integrating computer vision, robotics, machine learning and knowledge representation, and this will provide additional insights into the acting-humanly perspective. At the same time, several scientific and technological achievements, as I will introduce in the third section, have made it possible for us to claim that we are at the beginning of a new revolution in AI in which the goal of acting humanly will become a reality. The use of the same general models for language recognition and generation, imaging processing and speech conversion and production is a promising path in this direction.

The application of LLMs or foundational models endorses several of these properties in a very interesting way. However, I consider that there are also differences between acting humanly and AGI. Pennachin and Goertzel (2007) consider AGI more engineering-oriented than philosophical and scientific. In this sense, acting humanly has more room for philosophy, linguistics, and other research fields in which non-engineering analysis is still permitted.

3. Linguistics approaches relevant to AI

In this section, I will provide a brief and general context aiming at discussing, in the next section, the natural language processing perspective. Many of the theoretical results were obtained during the second part of the 20th century, particularly, by Chomsky's theories of language. Attribute grammars and Probabilistic linguistics also played a relevant role.

The analysis of languages considers several levels: the phonetic and phonological, the morphological analysis, the syntax and the discourse. The first level is the study of the different sounds that are part of the uninterrupted flow of spoken languages. In natural languages, sound production is constrained by the "phonetic apparatus" that is formed by organs (tongue, limbs, throat, nose, and alveolus) and structures (such as mouth cavity). Two groups of sounds can be distinguished in the most natural languages: consonants and vowels. There was a debate on the existence of natural languages without vowels (see the classical paper of Halle, 1960, on the Kabardian language). I think that now it is widely accepted that this language has two vowels. Vowels share a set of features, such as height, roundness and backness, while consonants are classified according to airstream mechanism, voicing contrast, the place of articulation and the manner of articulation. This forms the phonetics of the language.

The set of sounds associated with a specific language is further classified into units called "phonemes" according to the criteria of assimilation, dissimilation, insertion, deletion, and context independence of its occurrences. A phoneme, the minimal unit of sound that allows us to distinguish a word from another, could be formed by a set of sounds, called "allophones". One specific sound could be a phoneme and an allophone at the same time. For example, in Spanish, /m/ is a phoneme but, at the same time, an allophone of [n] by assimilation: [n] transforms to [m] before /p/ and /b/. The phonological component of a language is systematic and has some general properties that allow us to distinguish one language from others (see Hyman, 2014, for a brief account of different phonological systems). However, there are additional constraints in language production, such as force, sentence emphasis and language melodies (variations in pitch of voice), that affect but don't change the phonology of a language.

Phonological analysis plays a relevant role in several AI applications, as we will see in the next section. However, it is important to keep in mind that in many other applications of AI, specially, LLMs in which databases and datasheets are involved for training and for querying, the written language forms are the starting point of the lexicon analysis. It is still necessary to solve some critical issues to make it possible the transition directly from speech to higher levels of language analysis; issues such as the standardisation of data and other requirements for training neural networks for using LLMs, among others.

In this sense, the morphological, syntax and discourse analysis can be considered as a complete domain. Two general approaches could be mentioned: Bottom-up and Top-down. The first begins from the morphology to reach discourse. Usually, its general approach is compositional, it claims that the whole is equal to the sum of its parts and the rules used in forming the different components of the level of description. So, the morphological analysis aims to provide criteria for word formation, generation, recognition, and computer's lexical processing. Syntax aims to use morphological data to propose formalisms for sentence formation, and generation, and computer models for linguistic parsing. Finally, discourse aims to provide rules for discourse formation including, the different kinds of conjunctive adverbs, conditionals, and relativisation, among others.

Pure compositional approaches face some important problems. One of them relates to some kind of sentences, typically, the "donkey sentences" introduced by Peter T. Geach in which anaphoric references are involved. Examples such as: "Every farmer that owns a donkey beats it". The problem here concerns the scope of quantifiers to correctly assign the truth value to the sentence. The two main translations into first-order logic following the compositional strategy, fail to do it.

One of the solutions to this problem came from non-compositional analysis. For example, Hintikka and Kulas (1983), proposed a solution based on partially ordered quantifiers from their "Game-Theoretical Semantics" (second-order logic) in which these cases can be solved. It is a top-down approach. Not all top-down approaches are non-compositional. Descartes' analysis method is compositional. It proceeds from the general problem to its basic components (analysis), and from them, he reconstructs the general problem (by synthesis). This is relevant because, as we will see later, Chomsky's approach to morphology follows a top-down approach and is not compositional, but his general perspective on syntax is compositional.

In 1956, Noam Chomsky introduced a revolutionary perspective on natural language analysis. In his renowned article "Three Models for the Description of Language", introduced three different classes of algorithms according to their mathematical complexity or properties. According to him: "We explore multiple perspectives on linguistic structure to assess their capacity to offer grammars that are both straightforward and insightful, while ensuring they generate all and only the sentences that form part of the English language". So, his goal is to provide a recursive definition of "sentence" in a natural language. These three models are currently called "Chomsky's hierarchy". By a "recursive definition" we mean to find a procedure that allows us to derive any sentence of a language in such a way it explains the correctness structure of the sentence, and, in case of ambiguity, the set of different structures associated with those sentences. This requirement is very relevant in the evaluation of the theories proposed by Chomsky. In general, "production rules" is the name associated with the rules used to generate a grammar. The format of the rule is as follows: A ⤀ ω, where A is a non-terminal symbol starting point, and ω is a sequence of terminal and non-terminal symbols and ⤀ is the recursive symbol (zero or n application of the derivation rules), then, ⤀ is the generating symbol. So, this rule is read: "A generates ω".

The simplest class of algorithms are called regular expressions, regular grammars, or finite state automata, depending on the focus of interest. "Regular expressions" refer to sets and their algebraic manipulation rules. Regular expressions represent sets that have the following three properties: a) the empty word (empty set) is a regular expression; b) the set of letters of the alphabet are regular expressions and c) Regular expressions are closed under the operation of + (addition or union), • (set product) and * (concatenation of a finite number of words, including the empty word).

Regular grammars designate the class of grammars that can be obtained in the following ways: a) given a vocabulary Σ or terminal symbols (for example, ε, a, b,c,…, where ε is the empty word or string), b) given a set of Non-terminal symbols (A,B,…); c) the set of rules: X→ ь (where ь is a terminal symbol), and X→ ьX^1 (right ramification grammar rule) or X→ X^1ь (left ramification grammar rule).

Finally, finite state automata are built in terms of a finite set of states, a set of transitions between states, and a set of labels that identify the transition between states. For example, if we have two states, 1 and 2, a possible transition between 1 to 2, could be labelled by *a* (like *a* terminal symbol). As observed, transitions are functions of the form: δ (X, ь)= Y, where X and Y are states and ь is the labelling symbol.

What is relevant for us is that all these three classes are equivalent; they are different ways of representing the same set of expressions, but they differ from a computational perspective: Regular expressions are the algebraic representation of specific sets; regular grammar is generative class of grammar, and the finite state automata, the set of grammars or regular expressions accepted by these automata, so, finite state automata are regular expressions' acceptors.

Context-free grammars, context-free languages or push-down automata is the name of the second class of formalisms introduced by Chomsky. These formalisms are very useful for natural and computer languages. As in the case of Regular expressions, these three classes are equivalent, but refer to different perspectives: The first for grammars, the second for algebraic classes and the third for the class of sets accepted by pushdown automata. In general, the languages accepted by these automata are those that can be modelled by one pushdown. For example, $a^n b^n$ for n≥ 0, is a context-free language, but $a^n b^n c^n$ for n≥ 0, is not, because we need two pushdowns to generate this language. This specific language can be generated by a more general class of grammars called "Context-sensitive grammars". But this class of grammar loses an important property of Context-free grammars: its explanatory power, very appreciated by Chomsky. So, Chomsky tried to limit himself to more predictive formalisms.

Finally, the third class of models for the description of languages are known as "unrestrictive systems", because their rules don't have any restriction. This means that you can delete any sub-expression, to change its position or to insert a new sub-expression. As observed, context-sensitive grammars are a proper subset of unrestrictive systems, but they are the simplest of unrestricted systems. Generally, speaking, this class of grammars have the same properties as the general Turing machines, as shown in the relevant literature. What this means is that Turing machines can be used to express these three general classes.

Chomsky's hierarchy proved to be of great relevance to understanding the nature of computation, its algorithms, their limits, challenges, and further paradigm alternatives. For example, in linguistics and computer programming, finite state automata are relevant for capturing the nature of the lexicon and the reserved list of words of a programming language. However, it is not appropriate for capturing the nature of the syntax of a natural language or computer programming language. So, after Chomsky introduced this hierarchy, enormous progress was made in the development of computer programming languages. (see Zahar,1974; Hofcroft *et al.*, 2001; Martin, 2000).

Chomsky was very innovative in the field of linguistics, always keeping in mind the mathematical and computational properties of his theories. Starting with the publication of Syntactic Structures (1957) in which he introduced the phrase structure paradigm based on the second model referred above, context-free grammars. This approach was relevant for the formulation of different linguistic analysis theories. In 1965 (*Aspects of the Theory of Syntax*), he reformulated his theory introducing several novel features, among them: a) the distinction between deep structures (D-Structures) and surface structures (S-Structures); b) D-Structure are modelled by a context-free grammar; c) the introduction of the transformational component of the grammar that maps D-Structures onto S-Structures, and d) the phonological component as responding to the same distinction between D-Structures and S-Structures. This phonological approach provided new insights for the computational analysis of the lexicon component, as we will briefly discuss in the next section. It is a combination of a top-down approach and a compositional one.

In 1973, Peters and Ritchie proved that from a mathematical perspective, theories like that presented by Chomsky in *Aspects* (with two components) have the same properties as Turing Machines. In general, this means that some non-recursive sets are generated. It is not good, because Chomsky looked for recursive grammars. In the conclusions, Peters and Ritchie (1973) indicated:

> Thus we can justify the intuition of virtually all linguists that natural languages are recursive. This motivates the desire, as seen for example in (Ref. 1, footnote 37, p. 208), of transformational linguists to restrict deletions so that transformational languages are recursive. Although we have shown that the restrictions currently imposed on deletions do not accomplish this, our results guide research into this problem. (pp. 82-83)

However, one relevant problem was to find empirically motivated restrictions on deletions, and transformations, so that you can restrict the set of languages as recursive and, at the same time, maintain other linguistic properties such as explicative adequacy. Chomsky (1973) published "Conditions on transformations" as an effort to limit the expressive power of generative transformational grammars. Several conditions were proposed: on tensed sentences, on passive sentences, and on specified subjects, among others. These conditions attempted to solve another problem related to transformation: to find an empirical justification to limit the cycle of transformations needed to arrive at the S-Structure. It was expected that the number of transformations from the D-Structure to the S-Structure be as small as possible to maintain the theory's simplicity. However, in some cases the D-Structure can be obtained in only one step, for example in sentence passivation; in other cases, the number of transformations is significantly high. Consider for example the Spanish sentence: "Se lo regalé". It comes from a D-Structure like "yo regalar (past simple tense) el (libro) a (Pedro)". Several transformations should be applied to this sentence to obtain the corresponding S-Structure: past-tense transformation, deleting the subject, substitution of "lo" by "el libro" ("regalé lo a Pedro") and change of position of "lo" ("lo regalé a Pedro"), substitution of "le" by "Pedro", and advance of "le" before "lo" ("Le lo regale") and finally, substitution of "se" by "le" ("Se lo regalé"). This introduces a lot of complexity to the theory.

A second problem concerns the role of semantics in transformational generative grammar. This is associated with D-Structure. In some cases of ambiguity, it is natural to presuppose that semantics are involved in the analysis from S-Structure to the underlying sentences. Examples such as those discussed by Jaya Tarigan (2022) involve semantics. Consider the following:

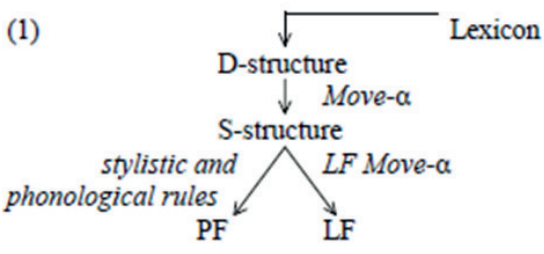
1. Lexical ambiguity: the visitors enjoyed the port.

2. Surface structure ambiguity: old men and women are advised to apply for their benefits

3. Deep structure ambiguity: cheating students will not be tolerated (Jaya Tarigan, 2022, p. 4)

Problems like these led Chomsky to formulate a new model called "Government and Binding" (GB). Direct reference to this model is observed in 1979 but the main developments took place in 1981, 1984 and 1986. Chomsky´s third theory incorporated several features of transformational generative grammars, but it introduces new features too. Chomsky schematised his theory in the following way:

Figure 1
Government and Binding Theory



Note. From Cheryl A. Black (1999, p. 2)

This new model is more intuitive and simpler, but at the same time, more abstract. It preserves the distinction between deep and surface structure and advances in the introduction of logical forms (semantics) into the model. From a computational perspective, the identification of constraints on Move-σ provides an interesting insight into how a natural language can be modelled, and at the same time, makes it feasible to translate one important significative part of the structure of one language into another. An interesting case is mentioned by Graf (2019) in which he suggests that "merge", a computational operation widely used in computer programming, is the inverse of "Move-σ", and both share other relevant features. However, my interest here is on the Natural Language Processing approach (NLP) that emerged from the two previous Chomsky´s theories of language.

In general, phonology is approached within Chomsky´s theories from a top-down perspective. This means that morpho-phonology is the most relevant level of approaching phonology. Since the sixties, when the first results from transformational generative grammars took place, suggested new insights on the relation between sentences and phonetics. In the recapitulation proposed by Chomsky & Halle (1968), they indicated:

> To recapitulate, a grammar contains a syntactic component which is a finite system of rules generating an infinite number of syntactic descriptions of sentences. Each such syntactic description contains a deep structure and a surface structure that is partially determined by the deep structure that underlies it. The semantic component of the grammar is a system of rules that assigns a semantic interpretation to each syntactic description, making essential reference to the deep structure and possibly taking into account certain aspects of surface structure as well. The phonological component of the grammar assigns a phonetic interpretation to the syntactic description, making reference only to properties of the surface structure, so far as we know. (pp. 6-7)

In this sense, the syntactic component is especially relevant for explaining the meaningful production of sound within language. Phonetics aspects such as sentence stress, variation in pitch and sentence emphasis should be explained by Grammar. So, the phonological component of a grammar receives as input the S-Structures applies a set of rules (transformation rules) to them and generates a phonetics sequence meaningful for a native language speaker. I will go back to this issue in the next section.

As mentioned above, other syntactic approaches to natural language are relevant for NLP. The first one is attribute grammar. It is a context-free grammar with attributes to any one of its nodes, including terminals. A node is every step in the derivation of an expression or sentence, according to the syntactic rules. One important property of these grammars is that attributes of a higher node are inherited by the lower nodes dominated by the higher node (parent), in a top-down approach. Or sibling-to-sibling attributes transmission in a left-to-right approach to analysis. This kind of grammar shares some important features with Object-Oriented programming. These attributes impose restrictions on the construction of some sentences on the selection of some lexicon words, or finally, conditions for forming complex sentences. Specific aspects of meaning features, and denotational semantics (for specific knowledge domains) could be integrated as attributes relevant to the derivation and explanation of the correctness of syntactic rules. In this sense, it provides a level of flexibility in the construction and analysis of a language.

Finally, probabilistic linguistics is a context-free grammar approach that incorporates frequencies (probabilities) of use to terminal and non-terminal symbols, and to morphemes to predict the most probable occurrence of some element in a sentence. For example, to a VP it follows a NP (transitive verb). Using this methodology in AI has made it possible to better tune predictions of the occurrence of a sentence or use some word with some specific meaning.

4. Natural language processing

In this section, I will indicate how the concepts introduced in the previous section are used from a computational perspective. This approach is called "natural language processing", and it is understood in the following way: "Natural Language Processing, or NLP, is a subset of artificial intelligence (AI) that focuses on the interaction between computers and humans through natural language. The objective of NLP is to read, decipher, understand, and make sense of the human language in a valuable way." (Kolena, 2024, p. 1). Other aspects of NLP include the conversion of images to text using OCR (Optical character recognition), very useful in digitalisation. It includes typewritten and handwritten documents. It remains, for example, many documents in archives that need to be digitalised and these conversion processes are doing the job (see Bhasin *et al.*, 2023; Ughade, 2024, for a brief account of this technology).

The speech-to-text conversion is another relevant application of NLP currently, that we have seen in different applications, such as cell phones, music lyrics, speeches, and movie subtitles. There are different methodologies to achieve it. The AI systems for speech-to-text conversion consist of several steps. One common element is the dictionary of phonemes (that includes, of course, its allophones). Personalised systems make this dictionary dynamic, that is, it tunes the dictionary to adapt the system to the voice of the user (variations in the pronunciation of individuals). But the dictionary also should be updated regularly, incorporating the changes that the spoken language undergoes. The recognition of meaningful features from the speech and associating with it a word or sentence requires the speech should be filtered before using it in the next step. Here morphophonological and syntactic information is relevant. On the other hand, the introduction of punctuation marks and special symbols still presents some challenges, especially in specialised domains.

The computational analysis of morphophonology was one of the successful results of NLP that goes back to 1987 when Kimmo Koskenniemi developed the first parser for morphological analysis. As indicated in the second section, finite state automata were the first class of algorithms introduced by Chomsky. This class is especially appropriate for lexical analysis (transducers). This class of algorithms are widely used in the construction of programming languages. Following Chomsky and Halle's insights on the transformational generative grammars for phonological analysis, Karttunen *et al.* (1987) implemented a two-level approach to morphology. The first level corresponds to the lexical strings, they exemplified it the lexical string "spy-s" formed by the noun root "spy", the morpheme boundary "-", and the plural morpheme "s". The second level corresponds to the surface strings that generate, in this example, the correct or legal correspondence "spies". The starting point of the morphological analyser was the written form of the language under consideration, so, it was not necessary to introduce morpheme to speech conversions. Lexical strings and surface strings are connected by rules. These were called "orthographic rules", and have the following structure: a quoted string, such as "Voicing rule 1", followed by a correspondence, an operator, any number of environments and variable assignment, if needed. For example,

"Voicing rule 1"
K:g ↔ vowel 1 ← vowel 2

This rule reads: "k is realized as g always and only between two vowels". Dalrymple *et al.* (1987) present a detailed account of this first computational account of morphology, in its first formulation.

Two aspects are relevant to remark on this approach. First, the lasting use of a two-level approach in the current development of lexical analysis. For example, "Tokenisation", "lemmatisation", "stemming" and "sentence segmentation" are current ways of programming how a machine must find words in large databases that match the prompt request. Let´s quote Jurafsky and Martin (2007) on the first two:

> English words are often separated from each other by whitespace, but whitespace is not always sufficient. New York and rock 'n' roll are sometimes treated as large words even though they contain spaces, while sometimes we´ll need to separate I'm into the two words I and am (tokenisation)
>
> Another part of text normalization is lemmatization, the task of determining that two words have the same root, despite their surface differences. For example, the words sang, sung, and sings are forms of the verb sing. The word sing is the common lemma of these words, and lemmatizer maps from all of these to sing. (pp. 4-5)

The second aspect that I would like to mention concerns one of the most interesting features of the chatbots and assistants: the automatic discovery of rules. Koskenniemi in 2006 describes how this automatic procedure could be introduced in the two-level morphological analysis to "discover" new morphological rules. His "underlying idea there was that one assumes a kind of distance metric among letters or phonemes as a computational formula" so "the sum of the distances between corresponding characters of the same morph would be minimized" (Koskenniemi, 2006, p. 426). The use of a geometric procedure was first implemented by Amazon in the late 1990s, and it is currently used in many applications and websites of AI, such as YouTube and searching in web navigators. So, it is very interesting to see how innovative the field of NLP at that time was.

However, there are other methods to make it more efficient for pre-training and answering queries on large databases. At the same time, it is observed that the terminology differs, in some authors, in scope. For example, Naveed *et al.* (2024) discuss different methods for training and processing queries. Naveed *et al.* (2024) use "tokenisation" in a more general sense than Jurafsky and Martin (2023). For Naveed *et al.* (2024), "tokens can be characters, subwords, symbols, or words, depending on the tokenization process. Some of the commonly used tokenization schemes in LLMs include wordpiece, byte pair encoding (BPE), and unigramLM" (p. 4).

Chomsky's hierarchy, introduced in the previous section, had played a relevant role in the development of computer languages since the 1960s. The programming languages distinguished three main parts: the lexical, the syntactic and the semantic component. The lexical component is modelled by finite state automata as introduced in the previous section. It is distinguished between "reserved words" and "non-reserved words". The first one refers to the set of expressions that have a special meaning in the language. This includes, for example, "if…then…else", "do…until", and assignment operators. The non-reserved words are defined as those that have a special meaning only in particular contexts and can be used as identifiers in other contexts. These two sets are complementary and accepted by different finite state automata. As mentioned above, in general, the lexical component of a natural language is analysed by finite state automata, by regular expressions, or by regular grammars. In this sense, both computer and natural language processing used the same class of algorithms.

The same is true in connection with the syntactic component: both computer programming and Chomsky's linguistic theories utilised Context-Free Grammars (CFGs) as formalisms. In this sense, both approaches are closely related. We will see the relevance of this in the next section. For a better understanding, a CFG is defined as G= ‹V,T,P,S›, where V is the non-terminal symbols or variables, T is the terminal symbols or constants, P is the set of productions or rules, and S is the starting symbols (S indicates "sentence"); the rules are of the form $A \rightarrow \sigma$, (σ derives from A). Chomsky used these kinds of grammars in his publication of 1957, *Syntactic Structure*.

From an algorithmic perspective, there are two ways of derivation called left derivation (if the terminals appear in this order: from left to right). The second one is right derivation (if the terminals appear from right to left). Chomsky used a left-to-right derivation. At the same time, he introduced one of the two more relevant normal forms for CFG. A normal form is defined as a model from which any other formulation of CFGs can be reduced. In the case of Chomsky's normal form (CNF), this general result is true only in the case in which the empty expression (ε derives only from S, not from intermediate variables). The rules of CNFs are of the following form:

$S \rightarrow \beta\gamma$ or $S \rightarrow a$, where S, β and γ are variables, and a is a constant.

As observed, Chomsky prefers to use only two variables on the right side of the rule (for example, $S \rightarrow NP+VP$, $NP \rightarrow det+N$). This makes that the derivation by trees reduce very significant the computation time. Four years later, in 1961 it was developed the CKY algorithm (Cocke–Younger–Kasami algorithm) for parsing Context-Free Grammars in Chomsky's normal form. It is an automatic procedure for deciding, given a sentence σ (with only terminal constants), and given a grammar G, if σ is $L(G)$ (if σ belongs to the sentences generated by the grammar G). It uses dynamics tables for determining if a sentence derives from S. CKY algorithm and CNF were relevant for the construction of computer programming languages and for investigations of NLP.

In 1965, Sheila Greibach proposed a second normal form, called Greibach normal form (GNF). Sheila Greibach was born in 1939 in the United States, then she arrived at this result when she was 25 years old. In GNF all the rules are of the following form: $S \rightarrow a\sigma$, where a is a terminal and σ is a sequence of (zero or more) non-terminals. As observed, it is a left-to-right derivation. GNFs are important for the construction of parsers. That is computer algorithms that accept Context-Free Grammars. The most relevant of GNF is that the parser proceeds from left to right reducing the number of attempts of derivations as it advances to the right and making the parsing process more efficient. On the importance of parsing, Jurafsky and Martin (2007) indicate:

> Syntactic parsing is the task of assigning a syntactic structure to a sentence. Parse trees […] can be used in applications such as grammar checking: sentences that cannot be parsed may have grammatical errors (or at least be hard to read). […]

> Parsers and the grammatical structure they assign a sentence are useful text analysis tools for text data science applications that require modelling the relationship of elements in sentences. (p. 367)

Several relevant applications derive from natural language processing. Among them, information extraction. During the 80´s and 90´s important computational efforts were made to extract information from orders written in different formats. Now, the conversion of unstructured information to structured one (for example converting unstructured information into standard databases) to be used for different purposes (training AI systems or querying information) showed the potential of NLP. Another application is the relation-extraction of different events, time, or template filling (for "recurring stereotypical events") (Jurafsky and Martin, 2023). This application could play a relevant role in teaching, in the analysis of information and social media.

In the semantic component, one interesting application is the thematic roles. These have been programmed to help the algorithm to make more precise claims and judgements according to prompts or querying. Grammar with attributes is the common theoretical perspective that can be used to do it: to embed in some lexical pieces (subjects, verbs, predicates, etc.) some semantic information that is used to make more precise answers or searches that the computer-programmed realized. Embedding is one of the main strategies used in AI for assigning probabilities to sentences, or verbal meanings to tune the most probable context use of them.

Jurafsky and Martin (2023) illustrate the use of thematic roles in NLP, with the following:

Table 2
Some thematic roles and their operational definition

Thematic Role	Definition
AGENT	The volitional causer of an event
EXPERIENCER	The experiencer of an event
FORCE	The non-volitional causer of the event
THEME	The participant most directly affected by an event
RESULT	The end product of an event
CONTENT	The proposition or content of a propositional event
INSTRUMENT	An instrument used in an event
BENEFICIARY	The beneficiary of an event
SOURCE	The origin of the object of a transfer event
GOAL	The destination of an object of a transfer event

Note. From Jurafsky and Martin (2023 p. 442)

As computer increases their processing capacities and storage capacities increase, more complex and interesting applications of natural language are possible. However, there is a discussion on the relevance of NLP in the new chatbots and virtual assistants, such as ChatGPT. This is the last topic that I want to address in this paper.

5. NLP and LLMs or Foundational Models

As indicated at the beginning of this paper, in 2022 we were surprised by the publication of ChatGPT-3, and we realised that a revolution in AI was taking place. In this section, I want to explore the connection between NLP and this new approach on AI. As indicated above, much of the technological achievements and theoretical perspectives in NLP come from the symbolic paradigm. As remembered, this paradigm assumed that,

> human cognition is analogous to symbolic computation in digital computers. On the classical account, information is represented by strings of symbols, just as we represent data in computer memory or on pieces of paper. […] The classicist believes that cognition resembles digital processing, where strings are produced in sequence according to the instructions of a (symbolic) program. (Buckner and Garson, 2019, section 5)

Our previous approach consisted of finding a set of rules, within a mathematical model with specific properties (CFG and Regular grammars), confronting them with linguistic data (coming from phonology, syntax, and semantics) to prove its robustness and, at the same time, constructing computer programmes in which these findings can be modelled and scaled. However, we remain within the traditional perspective of capturing intelligent capabilities in terms of explicit rules. A competing paradigm is associated with the radical changes observed in the new chatbots and virtual assistants. Then, to see the connection between NLP and these new developments, it is important first to introduce some general features of the new paradigm, and then analyse the relations between NLP and the foundational models (or LLMs).

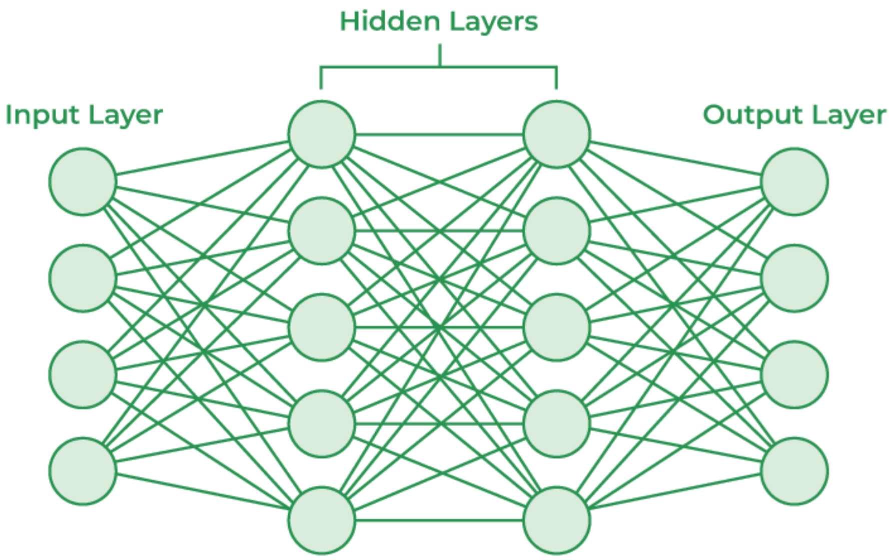
As mentioned in the first section, connectionism is the new AI paradigm and currently groups several research initiatives in AI, including computer science, cognitive science, economics, linguistics, philosophy, and physics. Like any other idea or scientific interdisciplinary field, it has a long history in science and philosophy. However, it was in 1943 that McCulloch–Pitts proposed the use of neural networks for modelling some perception features. This model was implemented by Rosenblatt in 1957. But it was in 1986 when David E. Rumelhart and James L. McClelland published two volumes titled Parallel Distributed Processing or PDP in which several theoretical models (architectures and propagating mechanisms) and applications were provided, all of them based on connections between artificial neurons. So, the field expanded faster after this publication. In general, these models are a subclass of “connectionism”. Connectionism has also provided insights for the formulation of an alternative theory of mind, very relevant in philosophy.

As summarised by Buckner and Garson (2019), a neural network could be understood as consisting of

> [...] large number of units joined together in a pattern of connections. Units in a net are usually segregated into three classes: input units, which receive information to be processed, output units where the results of the processing are found, and units in between called hidden units. If a neural net were to model the whole human nervous system, the input units would be analogous to the sensory neurons, the output units to the motor neurons, and the hidden units to all other neurons. (par. 3)

The following picture illustrates the three classes of units and the pattern of connections.

Figure 2
Neural Network (NN) sketch



Note. From https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/

At the beginning of PDP models, the process of training was very difficult not only on time but also on the availability of data for doing it and the limitations in computer capacities. The training of an NN consists of adjusting the initial values or weights of the NN, in such a way that the current output (C(O)) converges to the expected output (E(O)), that is, reducing the error range to a certain value. One of the algorithms for doing it is backpropagation, which consists of comparing the C(O) with E(O) and if they don´t converge to some specific value, then, some adjusts are made to the weights going from the C(O) through the hidden neuron to the input. The cycle is repeated till it converges to the expected output. The values assigned to each neuron after training are the set of weights of the ready state of NN; so, each neuron contributes to the solution of the proposed problem. To do it important computing power is needed. The current revolution in AI depends on the following three factors, as pointed out by Bommasani *et al.* (2021):

> The scale required three ingredients: (i) improvements in computer hardware — e.g., GPU throughput and memory have increased 10× over the last four years (ii) the development of the Transformer model architecture that leverages the parallelism of the hardware to train much more expressive models than before and (iii) the availability of much more training data. (p. 4)

Currently, NNs present an extraordinary complexity, involving hundreds of billions of weights that are adjusted during the training. After being trained these NNs are also adjusted minimally to fining the prompt for answering or tuning the NN to answer questions on some specific domains. Multilayer Networks is the name that sometimes they receive.

"Deep learning" and "machine learning" are concepts associated with AI. Machine learning is more general than deep learning; the last one is a proper subset of the first. Machine learning refers to "computers learning from data using algorithms to perform a task without being explicitly programmed", while deep learning is related to complex structures of algorithms in which NNs are involved.

Machine learning can be associated with both the symbolic paradigm and the sub-symbolic paradigm. However, deep learning is only associated with NNs. Deep learning is a proper subset of Machine learning. Machine learning usually works on structured data (like conventional databases), while deep learning allows us to deal with semi-structured or unstructured data, the most abundant data. Many of the applications of NLP are related to structured data. However, the new requirements on databases and datasets used for training NNs, make that these data be pre-processed in some specific ways, for example, to prevent using misinformation or documents protected by property rights during the training and further answers provided by NNs under request.

The kind of data used for training a NN is classified into two categories: supervised learning and self-supervised (unsupervised) learning. According to Greeks for Greeks, a leading platform for computer science resources, these two terms can be understood in the following way:

> Supervised learning, as the name indicates, has the presence of a supervisor as a teacher. Supervised learning is when we teach or train the machine using data that is well-labelled. Which means some data is already tagged with the correct answer. […] Unsupervised learning is a type of machine learning that learns from unlabeled data. This means that the data does not have any pre-existing labels or categories. The goal of unsupervised learning is to discover patterns and relationships in the data without any explicit guidance. (Greeks for Greeks, 2024, n.p.)

One of the most extraordinary successes in technological development is the construction of NNs based on self-supervised machine learning. To do that complex processes are involved, both in training and in answering queries. Among them, the activation functions. This is one of the fundamental mechanisms used to find correspondence elements of a sentence, the translation of sentences from one language to another, and so on, and is used to predict the probability of an output. Usually, these are expressed in terms of non-linear differentiation of trigonometrical functions, especially, sigmoid functions. For example, for translating one sentence, one process called encoder is involved in the activation functions. Several function values are associated with any input symbol. These values are embedded into the lexical unit and other grammar elements. This a very rich topological space in which mapping the changes in the weights of the neurons during the process of training or of answering, preserving consistency and predictability. The second fundamental mechanism used is called the "transformer model". These are called "Foundational models" as introduced in Bommasani *et al.* (2021). The model acquires context and meaning by analyzing the relationships within sequential data, such as the words in this sentence. Transformer models utilize a dynamic array of mathematical methods known as attention or self-attention to identify the intricate ways in which even distant elements in a sequence influence and rely on one another (Nvidia, 2024). These self-attention mechanisms are used to improve the performance of the system and focus on the relevant aspects of the query, based on probabilities. To calibrate the probabilities several techniques are used, among them, embedding meanings to each lexical item, as mentioned above. Initial probabilities are calibrated in the interaction with users, which provides relevant information during the process of prompting. So, embedding includes several kinds of information: 1) information on the lexical item and its play in grammar; 2) to introduce different meanings associated with lexical units and other grammar aspects; 3) to assign probabilities of use of some of these meanings based on user-system interactions and 4) the assignment of weight values associated with that lexical item. All these work together during the computational processes, making it possible to increase the predictability. As indicated, these embeddings are translated into numeral expressions by functions. So, a complex set of numbers represents a lexical unit or a sentence. The activation function, as observed, is a complex process, and this is conducted automatically by the virtual assistants currently available.

One of the amazing things about these LLMs (Large Language Models) is their capability to make that emerges almost many complete natural language structures (including morphological, syntactic and semantics components) from complex sets of weights and activation functions that keep consistency and predictability of the whole LLMs.

Bommasani *et al.* (2021) discuss the appropriateness of the term "LLM" for describing these new technological achievements, and introduce "Foundational Models" (FM) as a more adequate term for considering the wide range of current applications and foreseeable developments. They describe this change in the following way:

> We introduce the term foundation models to fill a void in describing the paradigm shift we are witnessing; we briefly recount some of our reasoning for this decision. Existing terms (e.g., pre-trained model, self-supervised model) partially capture the technical dimension of these models but fail to capture the significance of the paradigm shift in an accessible manner for those beyond machine learning. In particular, the foundation model designates a model class that is distinctive in their sociological impact and how they have conferred a broad shift in AI research and deployment. In contrast, forms of pretraining and self-supervision that technically foreshadowed foundation models fail to clarify the shift in practices we hope to highlight. (p. 6)

This "broad shift in AI research and deployment" refers to the multi-modal use of these models that go beyond natural language to include, for example, music composition, painting creations or sculpture AI creations. In this sense, a broader term helps better understand the wide range of applications. The second reason for using FMs is to indicate that "a foundation model is itself incomplete but serves as the common basis from which many task-specific models are built via adaptation" (Bommasani *et al.*, 2021, p. 7). Considering them "incomplete" indicates that it is possible to improve them in several ways: from a mathematical and computational perspective, for assuring more transparency on how the system behaves, and making them more adequate to social, ethical, and philosophical considerations.

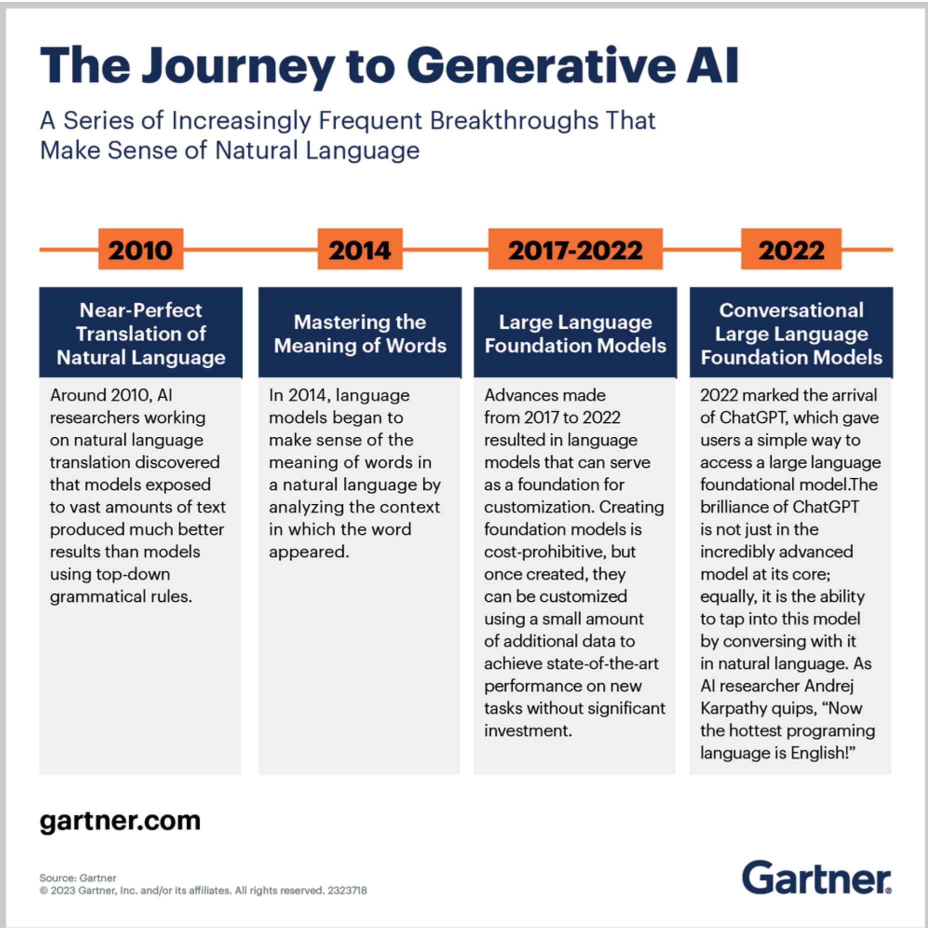
From the consideration of these models as foundational, Bommasani *et al.* (2021) associated two important properties with them: emergence and homogenisation. These are defined in the following way:

> Emergence means that the behaviour of a system is implicitly induced rather than explicitly constructed; it is both the source of scientific excitement and anxiety about unanticipated consequences. Homogenization indicates the consolidation of methodologies for building machine learning systems across a wide range of applications; it provides strong leverage towards many tasks but also creates single points of failure. (p. 3)

As mentioned above, emergence is a property of deep learning systems, while homogenisation indicates this observed tendency to abord different problems in the same way. In this sense, one of the most relevant methodologies is the use of category theory to consolidate the computational processes of NNs in terms of isomorphisms and homomorphisms. That is, mapping structures into structures. Yan (2023) proved some results (theorems) showing the promising of FMs as a general framework for understanding the current development and pointing out directions for future development.

It is clear from these considerations that NLP was an important step in the development of foundational models: it is in the evolution of FM. Gartner (2023), summarises the path from NLP to foundational models in the following way:

Figure 3
From NLP to Foundational Models



Note. From Gartner (2023, n.p.): https://www.gartner.com/en/topics/generative-ai

However, it is confusing the mix between Large Language and Foundational Models, because theoretically, they refer to different things as indicated above. Large Language Models (LLMs) are more specific models, while foundational models have the potential to be used in many other fields in which natural language is only one component. I think this is the case of Nvidia microchips that are strongly oriented to graphical information processes, they have an important NL component, but it is not reduceable to it. With these remarks, Gartner summarised well the process from NLP to foundational models.

6. Concluding remarks

From this presentation we may conclude the following:

a. Noam Chomsky is one of the most relevant intellectual figures in this story. Not only influence in the field of linguistics but also in computer science and in the intersection between natural language and computer science called Natural Language Processing (NLP).

b. Much of the work in NLP is framed with the symbolic paradigm, that is, whose results are expressed in the form of explicit rules as I emphasised in this paper.

c. In the current development of applications based on Foundational models (FMs), NLP played and is playing a relevant role, but differently: the language model (morpho-phonology, syntax, and semantics) emerges from the microprogramming of the Neural Network (NN), at the sub-symbolic level.

d. Therefore, there exists continuity between NLP and FMs as introduced in this paper.

References

Bhasin, K., Goel, A., Gupta, G., Kumar, S., and Bhoumick, A. (2023). *A secure mobile application for speech to text conversion using artificial intelligence techniques*. WINS/CVMLH.

Black, C. (1999). *A step-by-step introduction to the Government and Binding theory of syntax*. https://mexico.sil.org/sites/mexico/files/e002-introgb.pdf

Bommasani, R., Hudson, D., Adeli, E., Altman, R., Arora, S., von Arx, S., Berstein, M., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Quincy, J., Demszky, D.,... and Liang, P. (2021). *On the Opportunities and Risks of Foundation Models*. https://crfm.stanford.edu/report.html

Buckner, C., and Garson, J. (2019, May 18). Connectionism. (E. Zalta, Ed). The Stanford Encyclopedia of Philosophy. https://plato.stanford.edu/archives/fall2019/entries/connectionism/

Chomsky, N. (1956). Three Models for the Description of Language. *IEEE Transactions on Information Theory*, *2*(3), 113–124.

Chomsky, N. (1973). Conditions on Transformations. In S. Anderson and P. Kiparsky (Eds.), *A Festschrift for Morris Halle* (pp. 232-286). Holt, Rinehart & Winston.

Chomsky, N., and Halle, M. (Eds.) (1968). *The sound pattern of english*. The College of Ripon.

Dalrympe, M., Liakata, M., and Mackie, L. (2006). Tokenization and Morphological Analysis for Malagasy. *Computational Linguistics and Chinese Language Processing, 11*(4), 315-332. https://aclanthology.org/O06-5001.pdf

Dowling, J. (2018). *Understanding the Brain: From Cells to Behavior to Cognition*. M.W. Norton & Company.

Foothills Neurology. (2024). *Neurological Breakthroughs on the horizon: What to expect in 2024*. Retrieved August 14, 2024 from https://foothillsneurology.com/neurological-breakthroughs-upcoming-2024/

Gartner. (2023). *The Journey to AI*. Retrieved July 10, 2024 from https://www.gartner.com/en/topics/generative-ai

Graf, T. (2019). *Why movement comes for free once you have adjunction*. Stony Brook University.

Greeks for Greeks (2024) *Supervised and Unsupervised Learning*. Retrieved July, 10, 2024 from https://www.geeksforgeeks.org/supervised-unsupervised-learning/

Halle, M. (1970). Is Kabardian a Vowel-Less Language? *Foundations of Language*, *6*(1), 95-103. http://www.jstor.org/stable/25000430

Hintikka, J., and Kulas, J. (1983). *The Game of Language. Studies in Game-Theoretical Semantics and Its Applications*. D. Reidel Publishing Company, Dordrecht.

Hofcroft, J., Molwani, R., and Ulman, J. (2001). *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley.

Javaid, A. (2013). Brain-Computer Interface. SSRN Electronic Journal.

Jaya Tarigan, K. (2022). Illocutionary acts in Ernest Prakasa's movie "Imperfect". *Jurnel Pendidikan, 18*(2), 228-236.

Jurafsky, D., and Martin, J. (2007). Words & Transducers. En *Speech and Language Processing: An introduction to speech recognition, natural language processing, and computational linguistics* (pp. 1-42).

Karttunen, L., Koskenniemi, K., and Kaplan, R. (1987). *A compiler for Two-level Phonological Rules*. Stanford University.

Koskenniemi, K. (2006). *Multilingualism in the nordic and baltic area as a strength for language technology*. NordForsk.

Martin, J. (2000). *Introduction to Languages and the Theory of Computation*. McGraw-Hill. New York Fifth Edition.

Naveed, H., Ullah, A., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian A. (2024). *A Comprehensive Overview of Large Language Models*. *Arxiv*. https://arxiv.org/abs/2307.06435

Nvidia. (2024). *What is a Transformer Model?* Retrieved July 21, 2024 from https://blogs.nvidia.com/blog/what-is-a-transformer-model/ID

Pennachin, C., and Goertzel, B. (2007). Contemporary Approaches to Artificial General Intelligence. In B. Goertzel and C. Pennachin (Eds), *Artificial General Intelligence. Cognitive Technologies*. https://doi.org/10.1007/978-3-540-68677-4_1

Peters, J. R. and Ritchie, R. W. (1973). On the Generative Power of Transformational Grammars. *Information Sciences, 6*, 49-83.

Rescorla, M. (2024). The Language of Thought Hypothesis. In N. Zalta and U. Nodelman (Eds), *The Stanford Encyclopedia of Philosophy*. https://plato.stanford.edu/archives/sum2024/entries/language-thought/

Russell, S., and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc. https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf

Turing, A. (1950). Computing machinery and intelligence. *Mind*, (236), 433-460.

Ughade, N. (2024). *Decoding OCR Technology: A Catalyst For Streamlined Identity Verification*. https://hyperverge.co/blog/ocr-technology/#:~:text=OCR%20technology%20enables%20machines%20to,it%20faster%20and%20more%20accurate

Yan, Y. (2023). *On the Power of Foundation Models*. https://doi.org/10.48550/arXiv.2211.16327

Zahar, M. (1974). *Mathematical Theory of Computation*. McGraw-Hill.

Author's contribution

Pedro Salinas has participated in the data collection, research design, writing and critical review of the article and approves the version published in the journal of the article and approves the version published in the journal.

Acknowledgements

The author expresses his gratitude to the Review Team of this journal for its detailed review of this paper and for its valuable recommendations.

Funding

No funds were received for this investigation.

Conflict of interest

No conflict of interest.

Correspondence: celvargas@itcr.ac.cr

Author's academic career

Celso Vargas-Elizondo is Ph.D. in philosophy and full professor at the Costa Rica Institute of Technology, Costa Rica, at which he teaches. He is the author of more than 150 papers, chapters of books, and books. His academic interests include philosophy of science and technology, ethics of engineering, science and technology, and sustainable development. Recently, he has been engaged in understanding the complex phenomena of post-truth and their implications for democracy. He has a strong commitment against the new nuclear weapons race and proliferation, and in favour of the pacific uses of nuclear energy. He participates regularly in the activities of the CTBTO (Comprehensive Test-Ban Treaty Organisation), based in Vienna, Austria.