# The Genealogy of Large Language Models: From Auxiliary Tools in ASR to Foundational Transformers and Back Again

JOSÉ LUCIANO MALDONADO
Institute of Applied Statistics and Computing,
University of Los Andes,
Núcleo Liria, Av. Las Américas, Edif. G, Piso 1, Mérida,
VENEZUELA

*Abstract:* - This paper traces the evolutionary trajectory of Large Language Models (LLMs), arguing that their origins lie in the practical need to correct transcription errors in Automatic Speech Recognition (ASR) systems. We delineate this development, starting with domain-specific grammars, progressing through statistical n-gram models, and then to Artificial Neural Network-based models (ANNs), specifically RNNs, LSTMs, and GRUs, until reaching the pivotal breakthrough of the Transformer architecture. This evolution, driven by the pursuit of better language modeling, enabled the massive scaling that defines modern LLMs, which exhibit unprecedented capabilities. We conclude that LLMs, which emerged as an auxiliary component to mitigate the deficiencies of ASR systems, have "closed the circle" by becoming the foundational technology that now redefines the state of the art in their progenitor systems, thereby establishing themselves as a unifying technology for Artificial Intelligence.

*Key-Words:* - LLMs, Transformer Architecture, ASR, NMT, Language Models, RNN, LSTM, SMT, N-gram Models, AI History, NLP, Generative AI.

Received: March 8, 2026. Revised: April 3, 2026. Accepted: May 19, 2026. Published: July 27, 2026.

## 1 Introduction

Current LLMs are not an isolated technology that emerged solely to generate text and power chatbot conversations. The genealogy of their creation and development traces back to speech technologies, where ASR represents one of their key research fields. In other words, modern LLMs, which amaze with their ability to maintain conversations or generate coherent texts, did not emerge from nothing in an AI laboratory. Their technological roots lie in a practical problem shared almost simultaneously by two key fields: the difficulty of ASR systems in transcribing accurately and the challenge faced by Machine Translation (MT) systems in generating fluent text.

This paper does not seek to present new empirical results, but rather to offer a historical synthesis and a fresh perspective on the genealogy of LLMs.

The paper is structured as follows: The introduction (Section 1) presents the core thesis and the genealogical perspective of this work. We begin by describing the fundamental challenge of ASR that spurred the development of language models (Section 2) and detail the initial function of ASR systems as linguistic decoders (Section 3). Section 4 then establishes the state of the art circa 2000, contrasting progress in ASR and MT. Subsequent sections trace the revolutionary impact of key architectures: RNNs, LSTMs, and GRUs (Section 5), and the Transformer, which served as the pivotal breakthrough for LLMs and ultimately revolutionized their field of origin, ASR (Section 6). The quantitative impact of this evolution is assessed in Section 7, and the paper concludes with final remarks and future perspectives in Section 8.

## 2 Fundamental Challenge: The Imperfection of Acoustic Recognition

ASR aims to create technological tools with the capability to receive spoken messages and transcribe them into written text. To achieve this capability, an ASR system first receives the acoustic speech signal, converts it into a digital electrical signal, and then, from this signal, triggers a series of processes such as the detection of the phonemes present in said signal. Based on the detection of phonemes, other processes are triggered to identify which words are formed by

those phonemes, and subsequently, the system outputs the complete sequence of words present in the input signal.

In this process of identifying word sequences, errors in phoneme detection can occur, which may lead to errors in word identification. Consequently, it is not uncommon for the output sequence to have missing or extra words. That is, an ASR system can make various errors; such as inserting a word into the message, deleting words from the message, or substituting words in the message. These errors, which have different sources or origins, result in transcriptions that; in many cases; may lack meaning or be grammatically invalid in the language handled by the ASR system, [1], [2].

Therefore, since this is an inherent situation in ASR systems, researchers in this area turned to modeling the language of the operational contexts of ASR systems. It is noteworthy that initially, these were not referred to as LMs per se, but rather as context grammars or context-restricted grammars, as presented in the HTK toolkit manual, [3].

# 3 The First Solution: The Language Model as a Linguistic Decoder

The core of ASR evolution lies in its linguistic decoding subsystem, commonly known as the Language Model (LM), [1]. The primary function of this module is to resolve acoustic ambiguity by assigning a probability to a word sequence so that, among similar-sounding options, the system chooses the one that is linguistically most appropriate. For example, when faced with similar phonemes for the words “accept” and “except”, a well-trained LM would assign a higher probability to the phrase “it is an easy fact to accept” than to “it is an easy fact to except”.

The original objective of ASR systems has been surpassed, and systems now exist that not only transcribe text but also "interpret" the message and execute commands contained within it. For instance, they can perform database queries and present responses in spoken form as well. This means that ASR is now integrated into dialogue systems. The same occurs in MT systems, where the text output from ASR transcription can serve as the input to a translation system.

The point of this description is to highlight that ASR currently has diverse fields of application, largely due to the capabilities provided by the LM module they possess, [1]. In other words, the evolution of LMs not only improved transcription but also radically expanded the applications of ASR.

# 4 State of the art in ASR and MT around the year 2000

By the year 2000, the scope of ASR systems included recognizing isolated words, short messages with pauses between words (discrete speech recognition), and also continuous speech.

Isolated-word systems (such as those used for voice control) were very robust but limited. The major challenge and focus of research was continuous speech recognition (where words are pronounced consecutively, as in natural speech), which is much more complex due to coarticulation phenomena and fuzzy word boundaries.

The continuous speech ASR systems of that era were speaker-dependent, handled large vocabularies, but made many errors. They required a training or adaptation phase for the specific speaker. The user had to read a predefined text so the system could "learn" the characteristics of their voice (pitch, timbre; and accent).

Even under ideal conditions (adapted or dependent speaker, quiet environment, and high-quality microphone), word error rates (WER) easily reached 25% or more, depending on task complexity. In real-world conditions, these numbers worsened dramatically. The linguistic decoding subsystems worked based on n-grams, as rigid grammar-based decoding had already been surpassed because for complex tasks (such as dictation or news transcription), rigid grammars (e.g., regular or context-free grammars) that defined all possible phrases were unviable.

N-gram models became the standard for estimating the probability of a word sequence, allowing much more flexibility and coverage of natural language, albeit with limitations in capturing long-range dependencies. On the other hand, speaker-independent ASR systems existed, but their accuracy was significantly lower. Meanwhile, in the field of Machine Translation (MT), towards the late 1990s and early 2000s, the paradigm of Statistical Machine Translation (SMT) was dominant [4]. This approach represented a radical shift from previous rule-based methods, which required a costly and labor-intensive process of manual encoding of grammatical rules and dictionaries by human linguists.

The core of SMT was the Noisy Channel Model (NCM), a probabilistic approach from Information Theory, [5]. This model allowed framing the translation problem not as a linguistic rule-based problem, but as a decoding problem.

SMT faced the challenge known as the dichotomy between faithfulness and well-

formedness, [6], [7], where the Translation Model (faithfulness) was trained with enormous volumes of parallel text in two languages (for example, English and French), and learned the probabilities that a word or phrase in the source language, S, corresponded to a word or phrase in the target language, E. That is, it attempted to capture the translation model P(S/E). The major problems were ambiguity (since a word has multiple possible translations) and structural differences between languages, i.e., word reordering. Meanwhile, the Language Model (fluency), P(E), was essentially identical to that used in ASR, typically an n-gram model trained with massive amounts of monolingual text in the target language, to probabilistically measure word sequences in order to favor those that were common and sounded natural in the target language. However, the problem remained that the proposed translation often did not sound like a natural and grammatically correct text in the target language; that is, it had the same problems as ASR.

Of course, the noisy channel in SMT was not acoustic, but linguistic. The translation process alters a grammatically valid sentence E into an observed sentence S in another language. The decoder's task was to invert this corruption process. As can be seen, SMT was far from perfect. Its main limitations were the need for immense parallel corpora to train decent translation models, so for language pairs with limited resources, the results were very poor. Additionally, there was the reordering problem [8], with which simple phrase-based models (which succeeded word-based models) struggled with the different word orders between languages. The language model, P(E), attempted to correct the order, but generally failed in long sentences.

N-gram models, which could only capture local dependencies (usually between 3-5 words), were incapable of handling coherence at the sentence or paragraph level, leading to translations that might be fluent in short fragments but incoherent as a whole, or that lost gender, number, or tense agreement throughout the sentence. In general, the search for the highest probability resulted in literal translations or hallucinations, where the model chose a common but incorrect translation for an ambiguous word due to lack of context.

Therefore, any advancement in LMs would benefit both ASR and MT equally. Thus, the evolutionary pressure to create larger LMs, trained with more data and more efficient techniques, was driven by the need to improve the fluency of recognized speech and the fluency of machine translations.

Concretely, SMT and ASR shared a common problem, explained through the Noisy Channel Model paradigm, [9]. This implies that ASR assumes a valid word sequence W in the recognizer's application language passes through a noisy channel (the human vocal apparatus, air, microphone, telephone, or internet) to produce an acoustic signal A. The task is to find the most probable word sequence W that originated that signal A, i.e., $\overline{W} = \text{argmax}_W P(W/A)$. Using Bayes' theorem, this decomposes into the Acoustic Model P(A/W) and the Language Model P(W). Similarly, SMT assumes that a sentence E in the target language is the valid one, which passes through a noisy channel (the translation process) to produce a sentence in the source language S. The task is to find the most probable sentence $\overline{E} = \text{argmax}_E P(E/S)$, where P(S/E) is the Translation Model and P(E) is the Language Model.

It can be appreciated that, although ASR represents the primary and tangible practical use case, SMT and, later, Neural Machine Translation (NMT), were equally powerful research drivers for the development of more sophisticated LMs, especially in the era of RNNs/LSTMs.

## 5 Language Models and ANNs

In the early 2000s, even before the widespread adoption of RNNs, a pioneering approach emerged, Neural Language Models (NLMs). Although computationally expensive for their time, they demonstrated that Artificial Neural Networks (ANNs) could learn continuous, dense representations of words (embeddings) [10], overcoming the limitation of n-grams that treated words as discrete, unrelated symbols. This advancement, while not yet solving the problem of long-range dependencies, was crucial in paving the way for RNN-based models.

With the emergence of RNNs and, subsequently, LSTMs, a qualitative leap occurred in the construction of LMs, as they succeeded in modeling long-range dependencies in sequences. For this reason, the relevant contributions of these two technological tools are mentioned below.

First, it is worth mentioning the contribution of Williams and Zipser in their work "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks" [11], in which they propose a general and powerful RNN architecture and the Real-Time Recurrent Learning (RTRL) algorithm. This algorithm processes sequentially arriving inputs and operates indefinitely in discrete time, where no clear beginning or end of a sequence is assumed; that is, it

handles sequences of variable length. This technology constituted a milestone, since predecessor algorithms and contemporary alternatives, such as standard Backpropagation (BP) [12] and Backpropagation Through Time (BPTT) [12], were not applicable for real-time language modeling.

The RTRL algorithm, like the BP and BPTT algorithms, is also based on gradient descent but works in real-time and locally. That is, it adjusts the network's weights immediately after receiving each new input datum, not after a batch of inputs or a complete input sequence, and the gradient calculation for a specific time instant t depends only on information available at that same instant t or at very close instants. That work laid the perfect foundation for learning finite-state grammars and recognizing temporal patterns in input sequences; however, its key limitation was the vanishing/exploding gradient problem, which prevented them from learning long-term dependencies (for example, remembering the subject of a long sentence).

Secondly, the contribution of Hochreiter and Schmidhuber's LSTMs, described in the work "Long Short-Term Memory" [13], must be mentioned. LSTM networks revolutionized the field of LMs by overcoming the fundamental limitations of the RNNs of Williams and Zipser. Their main achievement was solving the vanishing/exploding gradient problem, which allowed them to learn long-range dependencies, a task at which their predecessor RNNs failed dramatically.

The key advances of LSTMs that drove the evolution of LMs are detailed below. During training with algorithms like BPTT or RTRL, in standard RNNs, the gradients propagated backwards in time tend to vanish (become very small) or explode (become very large) exponentially, [14]. This prevented the networks from learning from events that occurred many steps back in the sequence (generally more than 10 steps). LSTM networks introduced an innovative architecture that allows information and error to flow through time without degrading, maintaining a constant gradient; and thus avoiding the vanishing/exploding problem. This enables them to handle temporal dependencies of over 1000 steps, allowing language to be modeled with a much deeper contextual "understanding" by maintaining coherence in long texts. The major innovation of the LSTM architecture was the forget, input, and output gates. These gates allow the network to learn to control what information to keep in its long-term memory and what to discard, largely solving the problem of RNNs by enabling the modeling of much longer dependencies.

Additionally, the training algorithm for LSTMs, a variant of BPTT, has a computational complexity of O(1) per weight and time step in this architecture, compared to RTRL for standard RNNs; which has a computational complexity of $O(n^4)$, where n is the number of neurons, making it infeasible for large networks.

The success of LSTMs not only made them the standard architecture for sequence processing for many years in areas such as ASR, Natural Language Processing (NLP), and time series prediction, but also laid the conceptual foundations for subsequent architectures. The ideas of gating and selective control of information flow directly influenced the development of variants like Gated Recurrent Units (GRUs) [15] and, conceptually, the attention mechanisms that are fundamental in Transformers, the dominant architecture in current LMs.

LSTMs were not an incremental improvement but a paradigm shift that solved the critical limitations of RNNs, enabling machines to process and "understand" sequences in a much deeper and more contextual way, even though their sequential nature makes them computationally expensive and difficult to train, which limited their scalability. The adoption of LSTM-based models led to unprecedented reductions in WER for ASR tasks and substantial gains in Bilingual Evaluation Understudy (BLEU) scores for MT, consolidating the dominance of neural approaches.

## 6 The Turning Point: The Transformer and Closing the Circle

Despite the qualitative leap represented by LSTMs and GRUs, they have a major limitation: their sequential nature. By processing information one word (token) at a time, they cannot leverage the massive parallelism of modern Graphics Processing Units (GPUs). Therefore, training these models on massive text corpora is computationally infeasible. The evolution of language modeling needed a new paradigm that would break away from this recursiveness.

The turning point came in 2017 with the proposal put forward in the paper "Attention Is All You Need", [16]. This work introduced the "Transformer" architecture, which completely dispenses with recurrence and relies exclusively on a mechanism called "attention" (specifically, self-attention).

The self-attention mechanism allows the model to simultaneously weigh the importance of all words in the input sequence for each word it processes.

Unlike an LSTM, which must "remember" the subject of a long sentence through its recurrent state, a Transformer can establish a direct and instantaneous connection between a pronoun at the end of a paragraph and the subject it refers to at the beginning. This ability to capture long-range dependencies without the information degradation inherent in recurrence constituted a superlative advancement. However, this capability is not its greatest advantage; rather, it is parallelization.

Having no sequential dependencies, the Transformer architecture could, for the first time, leverage the massive parallelism of GPUs, enabling the training of models with previously unimaginable volumes of data and parameter counts.

Starting from the Transformer architecture, LMs ceased to be models with a few million parameters and became models with hundreds of billions of parameters. That is, they became the LLMs we know today, [17], [18]. With this super-scaling, LLMs exhibit capabilities that go beyond calculating the probability of the next few words in a sequence, as they can capture the context, semantics, and pragmatics of language, something that, for example, around the year 2000, LM-related research could only hint at.

Finally, it can be observed that the evolution of LMs closes a circle, as LLMs, direct descendants of the modest language models designed to correct errors in ASR, have returned to their field of origin to revolutionize it. A highly representative example of this is OpenAI's Whisper [18], which is not a traditional ASR system with separate acoustic and language models, but rather a single, end-to-end Transformer model trained on large volumes of audio data. It is capable of using the language model not only to perform unprecedented robust transcription in different languages and under adverse acoustic conditions but also to handle post-processing tasks such as correcting spelling, capitalizing words, and resolving semantic ambiguities that an n-gram could never capture.

## 7 Quantitative Evolution of Performance in ASR and MT

To quantify the impact of LM evolution, Table 1 shows key performance metrics for ASR and MT across the three main technological eras. It demonstrates that the transition from early LMs to recurrent neural models was not only conceptual but also translated into tangible and quantifiable improvements in the core tasks that drove their development. As summarized in Table 1, both WER in ASR and BLEU scores in MT experienced significant leaps with the adoption of LSTMs. However, the recurrent paradigm maintained a fundamental limitation for massive scaling. It was the Transformer architecture that, by resolving this bottleneck, enabled achieving the performance levels that define the current state of the art.

In the Statistical era, typical WER values are presented for ASR on complex continuous speech tasks (such as transcribing the "Wall Street Journal" corpus) around the year 2000. Commercial systems like Dragon NaturallySpeaking operated within this range, requiring speaker adaptation, [19]. For MT, the values shown are for English-French translation on the Europarl corpus, where phrase-based SMT models established the state of the art with these scores in the mid-2000s, [6].

Table 1. Quantitative Performance in LM Evolution

| **Technological Age** | **Statistics (c. 2000)** | **ANNs (c. 2010-2015)** | **Transformers/LLMs (c. 2017-Present)** |
|---|---|---|---|
| **Representative Architecture** | N-gram Models | LSTMs/GRUs/RNNs (Hybrid HMMs-NNs Models and Neural SMT) | Transformer Architecture (e.g., BERT for NLU, Whisper for ASR, T5/GPT for MT) |
| **ASR Performance (WER %)** | 20% - 35% WER (continuous speech, large vocabulary) | 10% - 18% (e.g., Deep Speech 1/2) | < 5% - 10% (e.g., Whisper on LibriSpeech) |
| **MT Performance (BLEU Score)** | 20 - 35 (Europarl EN->FR) | 30 - 40 (e.g., Google NMT 2016, EN→FR) | 40 - 45+ (e.g., large models on WMT) |
| **Key Observations** | Highly domain-dependent. Captures only local dependencies | They significantly outperform classical methods. Improvement in fluency and long-range context | Massive context learning capability. General-purpose models that specialize through fine-tuning |

*Source: created by the author*

In the era of transition from statistical to neural models, hybrid HMM-RNN/LSTM models, and later purely neural models like Baidu's Deep Speech 2, reduced WER by half in many cases [20]. For MT, work on English-French news translation established the new state of the art, [21].

In the Transformer/LLMs era, ASR systems based on Transformers, especially those trained on massive amounts of data like Whisper, achieved unprecedented robustness, even in noisy conditions and with multiple accents [18]. For MT, models have reached BLEU scores exceeding 40 in several translation directions, [21].

The three technological eras show that WER performance improved from 20-35% in the Statistical era, through 10-18% in the ANN era, to even below 5-10% in the Transformer era. Meanwhile, BLEU scores managed 20-35, 30-40, and over 45 in the three eras, respectively, for translations such as English-to-French.

## 8 Conclusions and Future Perspectives

This paper argues that the gestation of LLMs did not occur in text generation laboratories, but rather emerged from the practical struggle to solve fundamental problems in applied fields. The need to improve the accuracy of ASR systems served as the primary cradle, while the challenge of MT acted as a parallel and equally crucial development driver.

The evolutionary trajectory of LLMs followed a defined path: it began with restricted grammars, progressed through statistical n-gram models, took a qualitative leap with the introduction of RNNs/LSTMs/GRUs, and culminated in the paradigm shift brought about by the Transformer architecture. This evolution, driven by the pursuit of better language modeling, enabled the massive scaling that defines modern LLMs.

The result is a historical "closing of the circle." LLMs, descendants of those primitive language models designed as auxiliary components to mitigate ASR deficiencies, have evolved to become the foundational technology that now redefines the state of the art in their progenitor systems. Transcription systems like Whisper exemplify how LLMs, integrated into end-to-end architectures, provide ASR with unprecedented contextual understanding and robustness.

Beyond ASR, LLMs have emerged as a unifying technology for Artificial Intelligence. They constitute the basis of a new paradigm of general-purpose models that, through massive pre-training, perform diverse tasks ranging from text generation and machine translation to summarization, search, and sentiment analysis, without needing to be specifically designed for each one, thereby integrating previously disparate areas of NLP.

In essence, language models emerged to correct local errors and have now established themselves as the most powerful tool for that same task, but on a scale and with a depth of understanding that transcends the purely grammatical.

Looking to the future, the described genealogy suggests that future advances will not come from isolated models, but from their integration into multimodal systems capable of understanding and generating language, audio, and images, and even interacting with the physical world; systems that will collaborate naturally with people. The future, therefore, lies in hybrid systems that leverage the best of LLMs and other complementary technologies, thereby closing new circles in the continual evolution of artificial intelligence.

*Acknowledgement:*
The author wishes to thank the Institute of Applied Statistics and Computing, University of Los Andes, Venezuela, for its support during the development of this research.

*References:*

[1] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, Discrete-Time Processing of Speech Signals. New York, NY, USA: Macmillan, 1993. ISBN: 0023283017/

[2] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Stanford University Press, Stanford, CA, USA, 2025, [Online]. https://web.stanford.edu/~jurafsky/slp3/ (Accessed Date: October 7, 2025).

[3] The HTK Book (for HTK Version 3.4.1), Cambridge University Engineering Department, Cambridge, UK, 2009, [Online]. https://htk.eng.cam.ac.uk/download.shtml (Accessed Date: October 12, 2025).

[4] P. F. Brown, J. Cocke, S.A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P.S. Roossin, "A statistical approach to machine translation," Comput. Linguist., vol. 16, no. 2, pp. 79–85, Jun. 1990. https://dl.acm.org/doi/10.5555/92858.92860.

[5] C. E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., vol. 27, no. 3, pp. 379–423, Jul. 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[6] P. Koehn, Statistical Machine Translation, Cambridge University Press, Cambridge, UK, 2012. https://doi.org/10.1017/CBO9780511815829.

[7] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in Proc. 40th Annu.

Meet. Assoc. Comput. Linguist. (ACL), Philadelphia, PA, USA, 2002, pp. 295–302. DOI: 10.3115/1073083.1073133.

[8] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in Proc. 2003 Conf. North Amer. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. (NAACL-HLT), Edmonton, Canada, 2003, pp. 48–54. [Online]. https://aclanthology.org/N03-1017.pdf (Accessed Date: September 2, 2025).

[9] F. Jelinek, "Continuous speech recognition by statistical methods," Proc. IEEE, vol. 64, no. 4, pp. 532–556, Apr. 1976. DOI: 10.1109/PROC.1976.10159.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," J. Mach. Learn. Res., vol. 3, pp. 1137–1155, Mar. 2003. https://dl.acm.org/doi/10.5555/944919.944966.

[11] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," Neural Comput., vol. 1, no. 2, pp. 270–280, Jun. 1989. https://doi.org/10.1162/neco.1989.1.2.270.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, Oct. 1986. DOI: 10.1038/323533a0.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

[14] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE Trans. Neural Netw., vol. 5, no. 2, pp. 157–166, Mar. 1994. DOI: 10.1109/72.279181.

[15] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. (EMNLP), Doha, Qatar, 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need," in Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008. DOI: 10.48550/arXiv.1706.03762.

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, "Language models are few-shot learners," in Adv. Neural Inf. Process. Syst., 2020, pp. 1877–1901. DOI: 10.48550/arXiv.2005.14165.

[18] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in Proc. Int. Conf. Mach. Learn. (ICML), Hawaii, USA, 2023, pp. 28492–28504. https://doi.org/10.48550/arXiv.2212.04356.

[19] X. Huang, A. Acero, and H.-W. Hon, Spoken Language Processing. Upper Saddle River, NJ, USA: Prentice Hall, 2001. ISBN-13: 9780130226167.

[20] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. V. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan and Z. Zhu, "Deep Speech 2: End-to-end speech recognition in English and Mandarin," in Proc. 33rd Int. Conf. Mach. Learn. (ICML), New York, NY, USA, 2016, pp. 173–182. DOI: 10.48550/arXiv.1512.02595.

[21] Proc. Eighth Conf. Mach. Translat. (WMT). Singapore: Assoc. Comput. Linguist., 2023, [Online]. https://aclanthology.org/2023.wmt-1.pdf (Accessed Date: October 5, 2025).

**Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)**

José Luciano Maldonado was the sole author of this work, responsible for all aspects of the research.
José Luciano Maldonado is a Systems Engineer with an M.Sc. in Control Systems Engineering and a Ph.D. in Applied Sciences. He is a researcher at the Institute of Applied Statistics and Computing at the University of Los Andes, Venezuela, specializing in Artificial Intelligence (AI), with a focus on speech technologies (ASR, TTS) and Natural Language Processing (NLP), as well as Data Science.

**Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself**

No funding was received for conducting this study.

**Conflict of Interest**

The author has no conflicts of interest to declare.