Language Models: Automatic Speech Recognition and Semantic Analysis

Arega Mikayelyan
National Polytechnic University of Armenia
Yerevan, Armenia
e-mail: aregamikayelyan.tt055-2@polytechnic.am

Ani Manukyan
National Polytechnic University of Armenia
Yerevan, Armenia
e-mail: a.manukyan@polytechnic.am

***Abstract*—Recent advancements in language models (LMs) have significantly transformed two key domains in natural language processing (NLP): automatic speech recognition (ASR) and semantic analysis. ASR has evolved from statistical models, such as HMM-GMM frameworks to deep learning-based and transformer-driven architectures that achieve near-human transcription accuracy. In parallel, semantic analysis has shifted from shallow statistical approaches to deep contextual understanding enabled by transformer-based models such as BERT, GPT, and T5. This paper examines the evolution of ASR technologies, the rise of language models for semantic analysis, and their individual challenges and future directions. By examining both areas separately, we highlight their independent importance and discuss how ongoing research continues to push their boundaries.**

***Keywords*—Automatic Speech Recognition, Semantic Analysis, Transformers, Deep Learning, Natural Language Processing**

I. Introduction

Artificial intelligence has revolutionized language technologies, particularly in the fields of automatic speech recognition (ASR) and semantic analysis. ASR focuses on accurately converting spoken audio into text, while semantic analysis extracts meaning, context, and intent from textual data. Although both areas contribute to human-computer interaction, they remain distinct in their goals, methods, and challenges.

Historically, ASR began as a statistical problem of modeling acoustic signals and decoding words, whereas the semantic analysis was confined to text-based methods such as bag-of-words or n-gram models. With the emergence of deep neural networks and transformers, both domains have seen dramatic improvements. This paper discusses the technological evolution of ASR and semantic analysis separately, providing an overview of their progress, current state-of-the-art approaches, and future challenges.

The advent of transformer-based LMs (e.g., BERT, GPT, T5) and end-to-end neural ASR (e.g., Listen-Attend-Spell, RNN-Transducers) has blurred these boundaries. Pre-trained LMs, fine-tuned on domain-specific data, now excel at extracting nuanced semantics—from pragmatic inference to discourse coherence—by leveraging contextual embeddings. Simultaneously, architectures like OpenAI's Whisper and Google's SpeechT5 integrate speech-to-text and text-to-semantics within unified frameworks, using self-supervised learning on massive multimodal datasets (e.g., LibriSpeech, CommonVoice). These models jointly optimize acoustic, linguistic, and semantic representations, enabling robust performance in noisy environments and for diverse linguistic phenomena.

Despite these strides, significant challenges persist. Semantic gaps arise when ASR outputs misrepresent speaker intent due to homophones or ambiguous phrasing. Acoustic variability (e.g., accents, background noise) continues to degrade ASR accuracy, while computational inefficiency hinders real-time deployment. Moreover, biases in training data propagate into both semantic predictions and speech recognition, raising ethical concerns for equitable deployment.

II. Automatic Speech Recognition (ASR) Models

Automatic Speech Recognition (ASR) refers to the technology that converts spoken language into written text. It is a core component of many modern applications, including virtual assistants like Siri and Alexa, transcription services, voice-controlled interfaces, and real-time translation tools. Over the years, ASR models have evolved dramatically, moving from simple statistical approaches to powerful deep learning-based systems that can understand speech with near-human accuracy.

In the earliest ASR systems, speech recognition relied on Hidden Markov Models (HMMs) combined with Gaussian Mixture Models (GMMs). These models attempted to capture the temporal nature of speech and the statistical distribution of acoustic features. However, they required hand-crafted features, such as Mel-frequency cepstral coefficients (MFCCs), and could not effectively handle the complex variability in human speech, such as accents, noise, or different speaking rates. While HMM-GMM models were state-of-the-art for decades, their performance plateaued due to their limited ability to model long-term dependencies in audio signals.

https://doi.org/10.51408/csit2025_13

The advent of deep learning revolutionized ASR. Deep Neural Networks (DNNs) replaced GMMs for acoustic modeling, leading to substantial accuracy improvements. Later, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks became popular because they are well-suited for sequential data like speech. These models process audio frames in order and can remember previous states, allowing them to better capture temporal patterns. However, RNNs still faced challenges in training efficiency and struggled with very long sequences due to vanishing gradient problems.

A major leap came with the introduction of end-to-end ASR models, which directly map raw audio features to text without requiring complex intermediate components. Models such as Connectionist Temporal Classification (CTC) and sequence- to-sequence (seq2seq) architectures eliminated the need for separate acoustic, pronunciation, and language models. Instead, they learned the mapping jointly, simplifying the pipeline. Examples include DeepSpeech by Baidu and Listen,

Attend, and Spell (LAS) by Google, which use attention mechanisms to align audio frames with textual output.

More recently, Transformer-based ASR models have dominated the field. Transformers, with their self-attention mechanism, can capture long-range dependencies in speech more effectively than RNNs. Models like wav2vec 2.0 by Facebook AI and SpeechT5 by Microsoft leverage large-scale pretraining on unlabeled audio data to learn rich acoustic representations. These pretrained models can then be fine-tuned for specific ASR tasks with relatively small labeled datasets, achieving impressive accuracy even in noisy conditions.

Another trend in modern ASR is the integration of language models directly into the decoding process. By combining acoustic understanding with powerful pretrained language models, such as GPT-style architectures, ASR systems can produce more contextually coherent transcripts. For instance, when the audio is unclear, the model can rely on semantic context to choose the most likely word.

Today's ASR models are robust to different accents, background noise, and even code-switching between languages. They enable real-time transcription, multilingual speech recognition, and even domain-specific adaptation, such as recognizing medical or legal terminology. As research continues, we are moving toward multimodal models that can process both speech and text jointly, improving understanding of meaning beyond just transcription.

III. Semantic Analysis with Language Models

Semantic analysis interprets *meaning*, *context*, and *intent* in text, moving beyond syntax to model relationships between concepts, entities, and discourse. Language Models (LMs) are computational systems that learn linguistic probability distributions, enabling machines to extract nuanced semantics—from pragmatic inference to discourse coherence. Historically reliant on shallow methods (e.g., bag-of-words), semantic analysis has been revolutionized by deep neural networks and transformer architectures, which dynamically weight contextual elements to capture abstract meaning.

Evolution of Language Models for Semantic Analysis

1. *Statistical N-gram Models*

Concept: Predict words using frequency of fixed-length sequences (e.g., trigrams: *"the cat sat" → "on"*).

Limitations: Surface-level co-occurrence patterns only. Unable to capture long-range context or abstract relationships.

2. *Neural Language Models (Pre-Transformer)*

Map words to dense vectors in semantic space. Captures analogies (*king – man + woman = queen*) and word similarity. Processes text sequentially with memory retention.

Strength: Handle variable-length context for sentiment analysis.

Limitations: Struggle with long dependencies, slow training.

3. *Transformer-Based Contextual Embeddings*

Core innovation:

Self-attention dynamically weights all words in a sequence, enabling holistic context modeling.

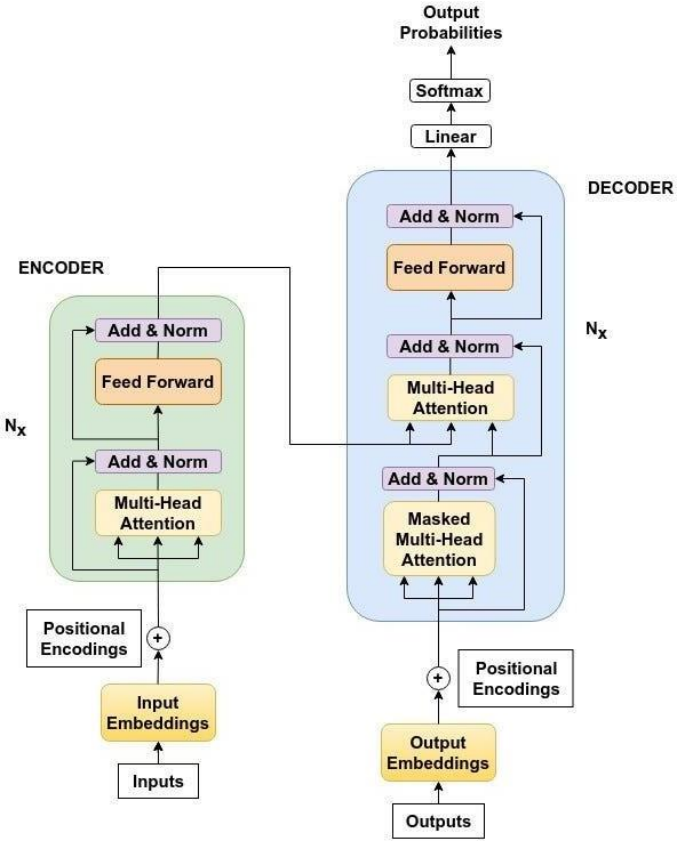

Figure 1. The transformer model architecture

The self-attention mechanism in transformers is a way for the model to weigh the importance of different words in a sequence when processing a specific word. It allows the model to understand relationships between words in a sentence, improving its ability to generate coherent and contextually relevant text. This mechanism is crucial for transformers, a neural network architecture that has become foundational for many modern large language models.

- Encoder models (BERT, RoBERTa)
- Decoder models (GPT, Llama)
- Encoder-Decoder models (T5, BART)

Fundamentally, both encoder- and decoder-style architectures use the same self-attention layers to encode word

tokens. However, the main difference is that encoders are designed to learn embeddings that can be used for various predictive modeling tasks, such as classification. In contrast, decoders are designed to generate new texts, for example, answering user queries.[4]

The original transformer architecture, which was developed for English-to-French and English-to-German language translation, utilized both an encoder and a decoder, as illustrated in Figure 2 below.[2]

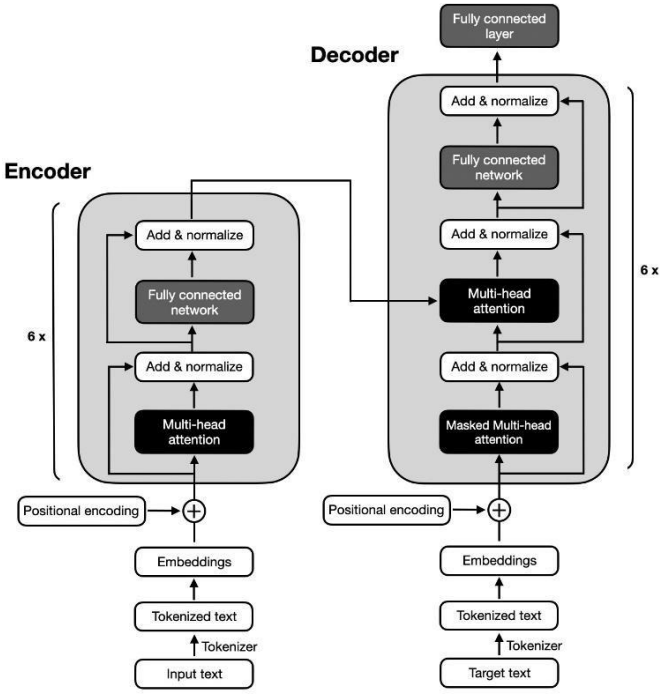

Figure 2. The Transformer architecture

The encoder in the original Transformer architecture, illustrated in the preceding figure, is responsible for understanding and extracting relevant information from the input text. It produces a continuous representation (embedding) of the input, which is then passed to the decoder. The decoder uses this representation to generate the output sequence, such as translated text in the target language.

BERT (Bidirectional Encoder Representations from Transformers) is an encoder-only model based on the Transformer's encoder module. It is pretrained on a large text corpus using masked language modeling (shown in the figure below) and next-sentence prediction tasks, enabling it to capture deep bidirectional context from text.[5]

In contrast, the GPT (Generative Pre-trained Transformer) series are decoder-only models pretrained on massive amounts of unsupervised text data and later fine-tuned for specific tasks, including text classification, sentiment analysis, question answering, and summarization. Models like GPT-2 and the more recent GPT-4 have achieved state-of-the-art performance on various NLP benchmarks and are among the most widely used architectures today.[4]

For tasks such as machine translation, the goal is to convert a source sequence (e.g., an English sentence) into a target sequence (e.g., a French sentence). The encoder-decoder Transformer is well-suited for this, as it efficiently models dependencies across both sequences.

For each word (token) in the sequence, we calculate:

- Query (Q): A vector representing the current word.
- Key (K): A vector representing each word in the sequence.
- Value (V): A vector carrying the information for each word.

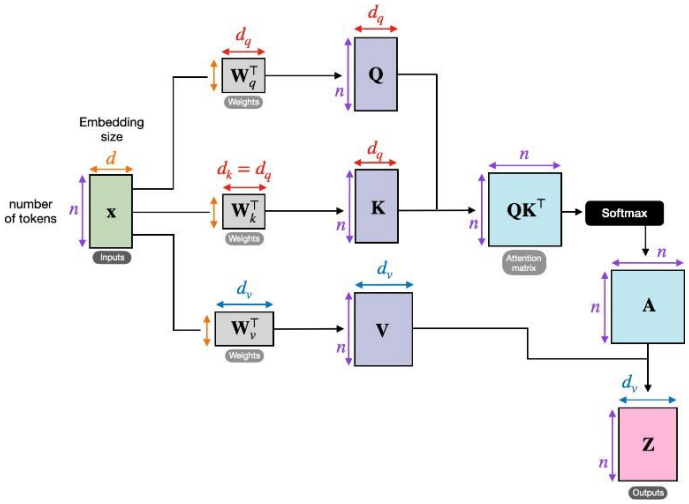

Figure 3. The Self-attention mechanism

The attention score for a given word is computed by taking the dot product of its query vector with all the keys in the sequence. This tells the model how much focus (attention) each word should have relative to the others.

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

$$\text{Attention Scores} = QK^T$$

$$\text{Scaled Scores} = \frac{QK^T}{\sqrt{d_k}}$$

This scaling helps prevent the gradients from becoming too large during backpropagation, mitigating the risk of the vanishing gradient problem.

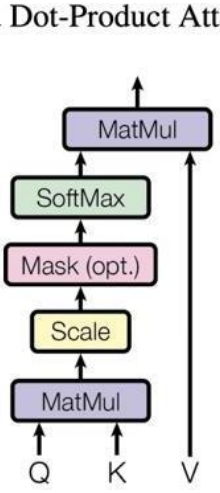

Figure 4. Scaled Dot-Product Attention

$$\text{Attention Weights} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

The softmax function ensures that all the attention weights sum to 1, making it easier to interpret these values as probabilities that dictate how much focus should be placed on each word in the sequence.

$$\text{Output} = \text{Attention Weights} \times V$$

This step results in a set of output vectors, each representing a word in the sequence but now enriched with contextual information from the entire input sequence.

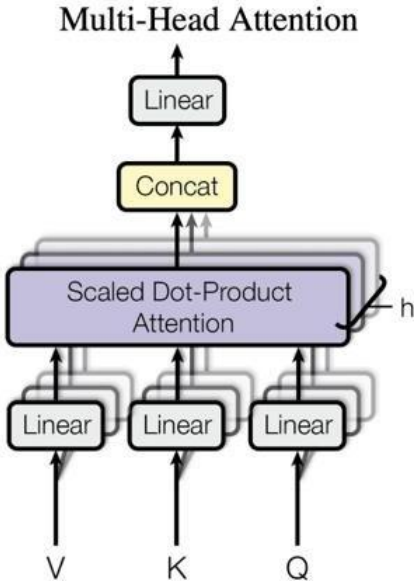

Figure 5. Multi-Head Attention

Multi-Head Attention is an advanced extension of the Self-Attention mechanism used within the Transformer architecture. This mechanism enhances the model's ability to focus on different parts of an input sequence simultaneously, thereby capturing a variety of perspectives and relationships within the data.

In essence, Multi-Head Attention involves repeating the self-attention process multiple times, with each repetition using different linear projections of the input data. This allows the model to attend to different aspects of the sequence in parallel, making the final representation more robust and contextually rich.

IV. CONCLUSION

The evolution of automatic speech recognition (ASR) and semantic analysis underscores a transformative journey driven by deep learning and transformer architectures. ASR has progressed from statistical models (e.g., HMM-GMM) reliant on handcrafted features to end-to- end neural systems (e.g., wav2vec 2.0, Whisper) that achieve near-human accuracy by leveraging self-supervised learning on massive datasets. Similarly, semantic analysis has shifted from shallow n-gram methods to contextual language models (e.g., BERT, GPT, T5), which dynamically interpret meaning through self-attention, resolving ambiguities and capturing discourse coherence.

Crucially, the boundaries between ASR and semantic analysis are blurring. Unified frameworks like SpeechT5 integrate speech-to-text and text-to-semantics pipelines, jointly optimizing acoustic and linguistic representations. This convergence mitigates traditional challenges such as semantic gaps (e.g., homophone errors) and acoustic variability (e.g., accents/noise) by using LMs to contextualize ASR outputs—enabling robust, human-like interaction in applications from virtual assistants to real-time translation.

REFERENCES

[1] H. M. Hesamian, W. Jia, X. He, and P. Kennedy, "Deep learning techniques for medical image segmentation: Achievements and challenges," *Journal of Digital Imaging*, Springer, New York, USA, vol. 32, no. 4, pp. 582–596, 2019.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., Red Hook, NY, USA, vol. 30, pp. 5998–6008, 2017.

[3] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, et al., "CodeBERT: A pre-trained model for programming and natural languages," *arXiv preprint*, arXiv:2002.08155, 2020.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., Red Hook, NY, USA, vol. 33, pp. 1877–1901, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Association for Computational Linguistics, Minneapolis, MN, USA, vol. 1, pp. 4171–4186, 2019.