A Review of Generative AI from Historical Perspectives

Preprint · February 2023

DOI: 10.36227/techrxiv.22097942

3 authors, including:

Dipankar Dasgupta
The University of Memphis
272 PUBLICATIONS **13,018** CITATIONS

SEE PROFILE

Kishor Datta Gupta
Clark Atlanta University
104 PUBLICATIONS **1,131** CITATIONS

SEE PROFILE

A Review of Generative AI from Historical Perspectives

Dipankar Dasgupta
Department of Computer Science
University of Memphis
Memphis, USA
dasgupta@memphis.edu

Deepak Venugopal
Department of Computer Science
University of Memphis
Memphis, USA
dvenugopal@memphis.edu

Kishor Datta Gupta
Department of Computer Science
Clark Atlanta University
Atlanta, USA
kgupta@cau.edu

***Abstract*—Many applications of Generative AI (such as DALL-E, GPT-3, ChatGPT, etc.) are making headline news in recent months and have been receiving both praise and criticism for their far reaching implications. Some of these applications include query responses, language translation, text to images and videos, composing stories, essays, creating arts and music, generating programs, etc. This review provides an historical background of Generative AI techniques and how they evolved over the years. This report highlights the benefits of Generative AI technologies and their limitations/challenges in moving forward. It is also to be noted that the large-scale applications of AI and their successes are now possible due to exponential advances in hardware (computational power, storage capacity), cloud computing and related operational layers of software.**

***Index Terms*—Machine Learning, Deep Neural Network, Evolutionary Computation, Language Learning Model, Transformer, etc.**

I. Introduction

"AI is not magic, it is logic"–Mathematics and statistics are underlying building blocks of all AI algorithms and require in-depth knowledge of techniques to efficiently solve different real-world problems [1]. There exists more than fifty so-called AI algorithms and heuristics. While some AI techniques are model-based, most others are data-driven and rely on large training datasets; also data quality and their properties (such as dimensionality, distributions, and 5-Vs of Bigdata [2]) generally affect classification, recognition, and prediction tasks.

For more than a decade, AI-based applications have been developed for several applications including controversial ones. For instance, image morphing (Face2Face, FaceApp) which can automatically modify someone's face to add a smile, make younger or older looking faces, or swap genders. Such apps can provide "beautifying" effects that include smoothing out wrinkles and, more controversially, lightening the skin. Such AI-based tools are also making it easy to generate realistic videos, and impersonate someone so that a person's facial expressions match those of someone being tracked using a depth-sensing camera [26]. Other AI tools (such as Lyrebird) are used to impersonate another person's voice and manipulate reality. Similarly, large language models (LLMs) have recently gained widespread popularity due to their ability to produce human-like answers to questions. At the same time, several questions remain about misinformation/disinformation that may be spread through the use of these models and the far-reaching impact of these models on future student learning and development.

All these technological transformations are possible due to hardware/software advances (such as massive computation power and exponential growth in storage capabilities). While AI-based tools are showing significant benefits, there is also an inherent danger that several so-called AI industry experts may use these powerful technologies as a *blackbox* without deep understanding of how they work and their inherent limitations.

II. Lessons from the past

Search and optimization are key to all AI algorithms, these use different similarity measures (pattern matching) to provide guided search in representation or problem space. To accomplish proper guiding, selection schemes, fitness function, loss function, transfer function, matching functions have been used to achieve to the desired goal. Most AI algorithms are probabilistic where the results need to be converted to decision or recommendation space. In general, performances of AI algorithms largely depend on tuning various control parameters (both internal and external), mapping functions, encoding schemes, distance measures, recognition thresholds, meta-heuristics, etc. Also for a specific application, hybridization of AI techniques, data pre-processing (sampling and dimensionality reduction) and post-processing (output filtering, and visual interpretation) play important roles in algorithmic success [1].

A. Evolutionary Computation

For last 50 years, Evolutionary Computation (EC) techniques (which include Genetic Algorithms, Evolutionary Algorithms, Evolution Strategies, Genetic Programming, etc.) have been applied in every discipline. ECs are population-based guided search which maps the problem (candidate solutions) in an encoded space. There are five major steps require in any EC model implementation:

- Representing a chromosome/individual as linear structure (bit strings, integer string, real-valued vectors, permutations, regular expression, etc.).
- Determining the fitness function which may need to be minimized or maximized based on the goal and objectives

of the problem (e.g. energy, loss, pollution, space, path, cost minimization; or profit, performance, efficiency, gain maximization) in pareto-front for multi-model problems.
- Selecting the parameters such as population size, generations (iterations), mixing operators (crossover, mutation, selection, replacement schemes, etc.) and other control parameters (elitist, niches,
- Performance measures and visualizing results and choosing terminating conditions (e.g., convergence criteria, maximum number of generations, best-so-far, etc.)
- Algorithmic implementation of ECs in computational platforms include parallelization (fine-grained, coarse-grained, hybrid), steady-state, distributed, clusters models, etc.

Over the years, ECs are used for design, scheduling, text mining, generate grammar rules, evolving neural networks, among others. Evolutionary Arts were developed by encoding the components (building blocks) in a chromosome/string and the population is evolved through human-computer interaction guided via subjective/aesthetic selection of user preferences. Karl Sims's GenArt media exhibition (such as Galápagos) allowed visitors to ”evolve” 3D animated creatures with different characteristics and capabilities [3]–[5].

Similarly, genetic evolution was used to build new solutions (design, music,..) from scratch, using component-based representations. Some of these applications involve handling non-linear constraints, multiple objectives, dynamic optimization, multi-modal problems, and others [6]–[8].

B. Genetic Programming (GP)

It is an evolutionary method for creating computer programs (in a functional language) from a collection of functions or code units to solve problems. It uses a bounded Tree-structured representation and cut-n-splice operators, among others. To improve the performance, GPs used Automatically Defined Functions (ADFs) to reuse code as libraries [1]. Many applications of GP are pursued including automated synthesis of analog electrical circuits, field-programmable analog arrays, controllers, antennas, etc.

C. Data Mining and Association rules

Association rule mining is a data mining technique that finds patterns and relationships among items in a given data. In association rule mining, we need to find frequent itemset first (data item that frequently occurs in a given dataset becomes a part of a frequent itemset). Association rule is defined as an implication of the form of if(antecedent) and then(consequent), which is written as $X \Rightarrow Y$. Apriori algorithm is the most basic rule mining algorithm that works based on prior knowledge of frequent itemsets [9]. A minimum support and confidence threshold is set for the algorithm before generating candidate itemsets. Other rule generation techniques such as FP-Growth algorithm [9], [10] appears to be faster and efficient in discovering notable relations and patterns at a multi-level through the Association rules components.

D. Artificial Neural Networks (ANN)

Works on ANN evolved from a single node perceptron to multi-layered fully-connected networks. For learning or training purposes, feed-forward and back-propagation algorithms were used to optimize the connection strengths (weights) as error minimization. There exist several neural network models such as Hopefield, Recurrent, Belief Networks. As learning algorithms supervised, unsupervised, semi-supervised, hebbian, reinforcement learning were being used. Also many variations of node transfer functions are used including simple threshold, sigmoid, ReLu, Radial Basis Function (RBF), etc. each having it's own activation properties. Convolution neural network (CNN) is an integrated neural network architecture with different segments of processing components which is able to perform complex machine translation, image/video and speech recognition tasks.

There exist other (symbolic and non-symbolic) AI techniques such as Fuzzy Logic, Cellular Automata, Immunological Computation, Tabu Search, Memetic Algorithms which have unique abilities in solving certain problems efficiently in resource-constrained environment.

III. Fundamentals of Generative AI

While some of the underlying ideas in Generative AI can be traced back to evolutionary computation models, the availability of big data along with exponentially more powerful hardware such as GPUs has been the driving force behind the significant impact that DNNs have had in this space. In particular, DNN based generative models have made remarkable progress in complex domains such as computer vision and natural language processing (NLP).

A. Generative AI in NLP

Generative models have revolutionized NLP and have achieved state-of-the-art results in several tasks. While traditional methods relied on knowledge-based approaches using linguistics, due to the availability of huge corpora of unlabeled text data, statistical/probabilistic generative language models have been the dominant approach over the past several years. Specifically, in generative language modeling, DNNs are trained to assign probabilities to the next word/phrase in a sequence of words in text, thus allowing us to generate new text. Some key NLP advances made by generative language models include question answering, composing essays/stories from prompts, sentence completion, reading comprehension, Natural Language Inference (where we infer relationships between sentences) or Machine Translation. Based on generative language models, algorithms such as BERT and Elmo were breakthroughs in NLP and have been applied in various tasks including understanding and answering questions [11]–[13]. Similarly, the OpenAI research group has demonstrated through their highly-publicized GPT algorithms [14] that generative language models can write almost human-like text with minimal prompts. At the same time, we need to be cautious about the usage of such models since the same algorithms can indeed be used maliciously, for example, to compose

[1] httpwww.genetic-programming.org/

authentic-looking fake news articles from a few pieces of information about the intended story [15]. Thus, it seems like regulating the use of such models is a critical need of the hour. Next, we provide an overview of some seminal DNN-based generative models, namely, Variational Autoencoders (VAEs) [16], Generative Adversarial Networks (GANs) [17] and transformers [18].

B. Generative AI DNN Architectures

VAEs are autoencoders that are trained to encode a latent representation of the input which when decoded reconstructs the input. The main idea is to restrict the latent vectors such that they are drawn from a Gaussian distribution. The training is formulated as variational inference where the encoder also called as the *inference network* is a neural network with parameters θ that takes as input x and outputs the parameters of the Gaussian that represents the latent vectors with probability $p_\theta(z|x)$. The decoder which is called as the *generative network* is a neural network with parameters ϕ that samples a latent vector from the Gaussian parameters and reconstructs the output $\hat{x}$ with probability $p_\phi(\hat{x}|z)$. The encoder and the decoder networks are trained jointly using backpropagation. Once trained, the generative network (decoder) can be used to generate new data. While VAEs are most widely-used for image generation, they can be used to generate different types of data including music, text, voice, etc.

Unlike VAEs that optimize a likelihood function, GANs are called *likelihood-free* generative models. That is, they do not optimize a likelihood function and in some sense this makes them less restrictive. The use of Gaussians in VAEs tend to make generated images with reduced clarity (e.g. they appear blurry). GANs on the other hand can generate more realistic images compared to VAEs but at the same time, they are much slower to train and do not converge quickly. The idea in GANs is again to have two neural networks, where one acts as a *discriminator* and the other acts as a *generator*. The discriminator optimizes a function where it can identify if data is generated by the generator or is the real training data. The generator on the other hand optimizes itself to generate images that are hard to discriminate with real-images. This yields a min-max training objective where optimizing the generator hurts the discriminator objective and vice-versa. The discriminator and generator are trained jointly using backpropagation and once the training converges, we can use the generator to generate new data instances. Like VAEs, through GANs, we can generate any form of data including images, text, voice, graphs, etc.

At the heart of state-of-the-art generative AI models for NLP (such as ChatGPT) are large-language models (LLMs) that use transformer architecture which consists of sequence of deep learning models that can produce a sequence of text (or perform language translation) for given inputs. Specifically, the transformer architecture introduced in [18] revolutionized language processing through the use of *attention mechanisms*. Transformers are now considered as the state-of-the-art in representation learning for Natural Language Processing (NLP).

Two state-of-the-art models developed using transformers include Bidirectional Encoders Representations from Transformers (BERT) [19] and GPT [14], [20]. The BERT model developed by Google has achieved state-of-the-art results in several challenging NLP tasks such as question answering, Natural Language Inference and Machine Translation. OpenAI's GPT models also use transformers and in particular, a multi-layer decoder transformer with multi-head attention.

The GPT model consists of multi-layer decoder transformer blocks that generate a new word based on words seen previously in the context window. Thus, the training is a form of *autoregression* and more specifically, it maximizes the following likelihood function.

$$L(\mathcal{U}) = P(u_i|u_{i-1}, \dots u_i, \Theta) \tag{1}$$

where the input $\mathcal{U}$ is a sequence of unlabeled tokens $u_1 \dots u_n$ and Θ denotes the parameters of the model. The model uses the masked self-attention mechanism where a token is allowed to attend to tokens prior to it in the sequence (unlike full self-attention where a token can attend to tokens that occur after it in a text sequence). The attention mechanism allows the model to understand the relative importance of different parts of the sequence and use this to generate the next word in the sequence. For example, for the prompt *a girl is wearing a red dress and playing with* $\dots$, for the next word in the sequence, the action of playing is more important than the color of the dress the girl is wearing. Multi-head attention simply computes this attention several times in parallel, thus allowing sub-sequences of text to be attended to differently giving it more flexibility.

Mathematically, the attention mechanism in transformers can be expressed in the form of the following equation.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where Q is called the query matrix, K is the key matrix, V is the value matrix and d_k is the dimensionality of the embedding that represents the latent representations. The query, key and value matrices are abstractions, where query can be viewed as an "input" (tokens that are seeking the attention), and (key,value) pairs refer to the "output" (which token has information about the attention) and how much attention each token has is denoted by the value which is encoded in the form of attention weights. The transformer architecture computes multiple attentions in parallel using multiple heads as given by the following equation.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{3}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$ and W_i^Q, W_i^K, W_i^V are projection matrices, and W^O is a weight matrix.

C. Pre-training Generative Models

A key step that is critical to the remarkable performance of GPT is *generative pre-trained transformer*. Specifically, the transformer model is pre-trained over extremely large and diverse corpora of unlabeled text. For example, the third generation GPT (called GPT-3) [21] is trained over data obtained from crawling the web, books, Wikipedia, etc. This allows the model to learn about language structure in general. The scale of the GPT-3 model is quite staggering at around 175 billion parameters. Thus, pre-training over large datasets allows the model to fit its parameters over general patterns seen in diverse contexts. Typically after pre-training, the model can then be fine-tuned for a variety of different tasks without the need for task-specific architectures and with supervision from a much smaller set of labeled examples. However, interestingly GPT-3 does not use fine-tuning. In particular, it is seen that fine-tuning hurts performance when we want to generalize to out-of-distribution test data. The goal of GPT-3 is to be task-agnostic and generalize to *few-shot*, *one-shot* and *zero-shot* cases.

In the few-shot case, the model is shown a few demonstrations of an NLP task and then given a novel instance to solve. For example, K instances of translation from one language to another are shown to the model and the model has to translate a new instance. In the one-shot case, a natural language description of the task is provided along with a single demonstration. This is also closely related to how humans learn, i.e., without requiring many cases for the same task. In zero-shot, no examples are shown and only the natural language description of the task is given to the model. The model is then expected to solve the task for a new instance. Arguably, this is the hardest and almost equivalent to how humans are expected to perform tasks. schematic view of GPT-3 is shown in Fig. 1. It has been shown that in all these settings GPT-3 (and later versions used in ChatGPT) have performed exceptionally well, thus pushing us closer towards general intelligence. In particular, the success of ChatGPT in generating responses from an initial prompt where the quality of text generated is virtually indistinguishable from human-generated responses has significant technological, social and ethical implications [22]. Next, we present a conceptual *pipeline* for generative AI applications.

D. Generative AI Pipeline

1) Preprocessing: Inputs to a generative AI pipeline may be homogeneous (a single type) or heterogeneous (a mixture of several types). Input types include numerical data, categorical data, text, voice, image or multimodal data to name a few. Data may be structured (e.g. tabular data) or more commonly in generative AI models, the data is typically unstructured. For instance, a study by MIT's Sloan school estimates tat 80-90% of the data in the world is unstructured. Depending on the type of unstructured data, one or more steps of pre-processing may be needed to load the data which in the case of large models is non-trivial. In particular, in the case of language models such as GPT-3, the CommonCrawl dataset used to train the model has a raw size of 45TB before pre-processing. After filtering, the data is reduced to 570GB. For pre-processing of this magnitude, commonly called ETL (Extract, Transform and Load) big data tools are utilized (e.g. Cloudera, Apache Spark, AWS, Google Cloud, etc.).

2) Tokenization: Tokenization converts unstructured data to structured data. The type of tokenization is specific to the data. In NLP, there is a rich source of tools to perform tokenization such as Stanford's NLTK. More advanced tokenization include converting between data types. In Speech2Text [23], text tokens are generated from speech signals. In the case of multimodal applications containing text along with other input types (e.g. images/videos), we need to perform alignment of tokens where text tokens are filtered/aligned with features in the image/video.

3) Creating Object Libraries: Language models form the core of text-based generative AI pipelines and map text to generated objects/components. The representation of these components is typically not highly interpretable in the case of DNN models. Specifically, DNNs as is well-known, use a sub-symbolic distributed representation where the information is distributed across the inter-connected units in the DNN. Thus, symbolic concepts (e.g. words) in text are embedded as vectors in the DNN. Importantly, the *embedding* contains semantic information, i.e., concepts that are close to each other in the real-world are close to each other in the embedding-space. Embeddings that are pre-trained allow downstream tasks to transfer/reuse DNNs without even without having access to the original (possibly very large) datasets that generated these embeddings. Thus, we can view embeddings as a library of objects that can be utilized by multiple downstream tasks. Note that to use these objects in downstream tasks, these embeddings need to be converted into human-interpretable form. In text-based models, a typical way to do this is to sample output tokens (such as words) based on the embedding based on a probability distribution. However, in such generation, we may need some form of external knowledge to ensure that the generation is meaningful. For example, a question answering system that teaches students needs to know concepts of physical laws of the world such as gravity, motion, etc. For a system that generates computer code, we need to store syntax and semantic information of programming languages. Encoding such commonsense knowledge into DNN libraries is a challenging task and one that is a subject of ongoing research. While pre-training with very large datasets can alleviate some of these challenges (assuming that commonsense knowledge can be extracted as patterns from large datasets), this is still recognized as a significant limitation in generative AI.

4) Downstream Tasks: Common downstream tasks involving generative AI include question-answering, Machine Translation, reading comprehension, Natural language Inference, language completion (sentence completion, paragraph writing, etc.). Both BERT and GPT models can be used in most of these downstream tasks. In multimodal settings that bridge language and visual representations, downstream tasks include text to image generation using methods such as stable diffusion.

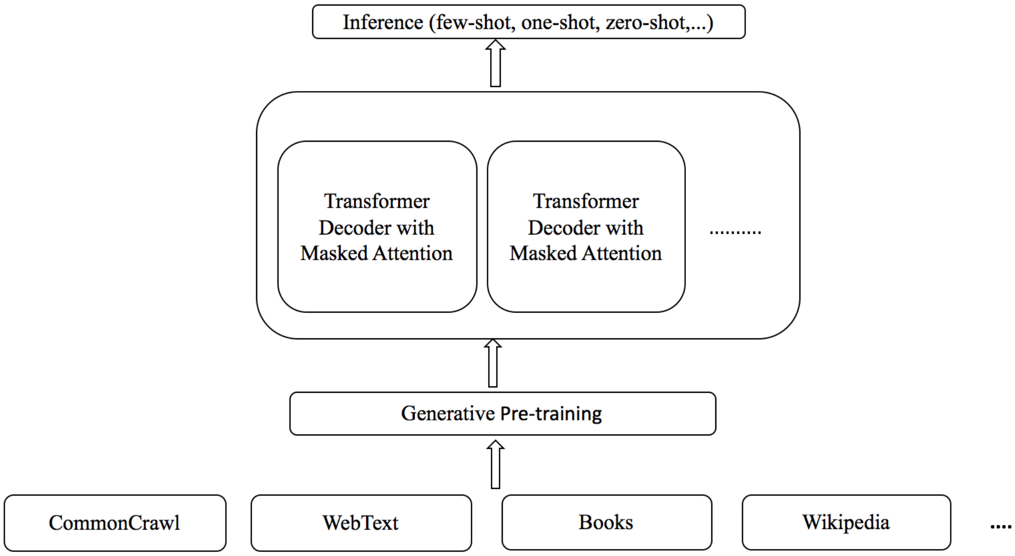
Inference (few-shot, one-shot, zero-shot,...)
Transformer Decoder with Masked Attention
Transformer Decoder with Masked Attention
....
Generative Pre-training
CommonCrawl
WebText
Books
Wikipedia
....

Fig. 1: A Schematic View of GPT-3.

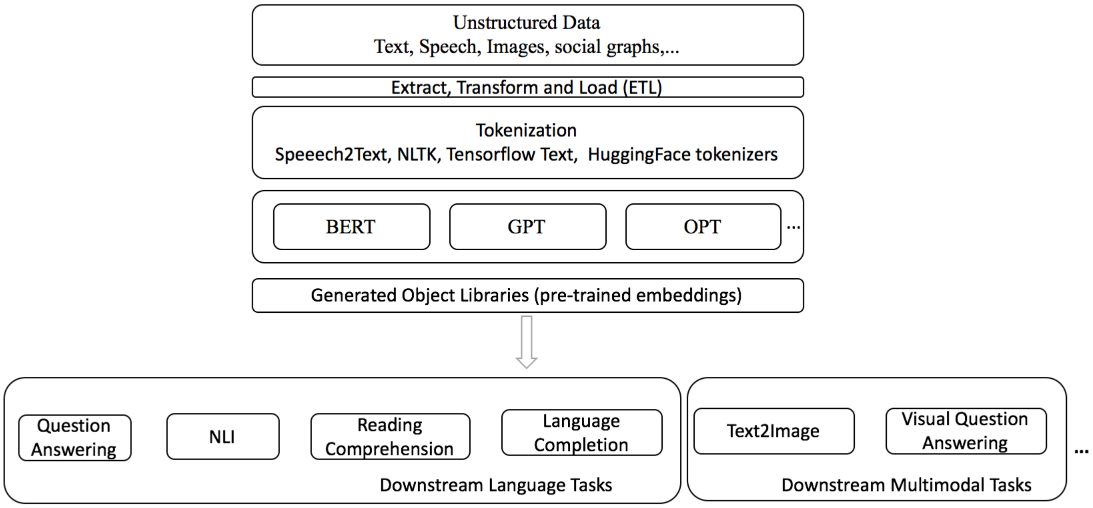
Unstructured Data
Text, Speech, Images, social graphs,...
Extract, Transform and Load (ETL)
Tokenization
Speeech2Text, NLTK, Tensorflow Text, HuggingFace tokenizers
BERT
GPT
OPT
...
Generated Object Libraries (pre-trained embeddings)
Question Answering
NLI
Reading Comprehension
Language Completion
Downstream Language Tasks
Text2Image
Visual Question Answering
Downstream Multimodal Tasks
...

Fig. 2: A Pipeline for Generative AI

Several other downstream applications and tools are described in the next section.

IV. Generative AI Applications

The discussion in this section highlights different products and services in generative AI that have transformed the way we live, work and access/consume information in general [24], [25].

A. Computer Vision

AI-based services for computer vision have revolutionized the way we approach image and video generation. Wide spread use of imagery is possible due to improved technology in producing high-quality photography and their availability to users. Storing and retrieving images in digital form allowed the search space to be bounded (even though very large) and this has helped researchers utilize AI techniques for pattern recognition and, search and optimization applications.

NightCafe Studio's AI art generation technology uses state-of-the-art DNNs such as GANs, to produce new and unique pieces of art. The technology incorporates a variant of the VQ-VAE-2 model [26] for generative art named VQGAN+CLIP [27], which uses a combination of Variational Autoencoder (VAE) and Vector Quantization (VQ) models to generate high-quality images. The VQGAN+CLIP model which incorporates the CLIP (Contrastive Language-Image Pre-training) algorithm [26] to improve the image generation capabilities by providing a better understanding of the relationships between different objects and concepts in the images. Wombo Dream is another popular app for art generation, developed using the similar technique VQGAN+CLIP. In particular, VQGAN generate images that look similar to other images, while CLIP is trained to determine how well a text description fits an image. The two algorithms work together in a feedback loop, with CLIP providing feedback to VQGAN on how to match the image to the text prompt and VQGAN adjust the image accordingly. The process is repeated thousands of times, resulting in a generated image as per the text description. Starryai is another art generator app that allows users to create unique and stunning works of art by simply entering a text prompt. The app was developed with the goal of making AI art more accessible to non-coders and aspiring creatives who may not have the technical expertise to create their own AI-generated art. The app uses two AI models to generate art, Altair and Orion. Altair uses the VQGAN-CLIP [27], [28] model to render the artwork creations. Orion, on the other hand, uses the CLIP-Guided Diffusion technique to create stunning artworks and imageries. The Diffusion component mathematically removes noise from an image, while the CLIP component is used to label images. When the two interact, CLIP iteratively guides the diffusion de-noising process through proper image identification, allowing the image to be created in alignment with the text prompt. The process starts with a very blurry image and gradually becomes more detailed and coherent as it goes through the iteration process. Among the art generation services, Midjouney is arguably the most famous company to create art from text prompts and sample art.

DeepVideo is a video generation platform that uses deep learning to generate videos from text and images. It is a valuable tool for creating videos for social media, explainer videos, and more. Autodessys is another AI-based 3D animation software that can create 3D characters and animations by using real humans as references. In the domain of video creation, Wibbitz is another AI-powered video creation platform that can automatically generate videos from written content. It uses natural language processing to understand the meaning of the text and generates a video with corresponding images and text overlays. Synthesia is another AI-based video generation service that can create videos by animating text and images in a variety of styles. Other AI-based services in the computer vision domain include Flair, which allows users to design branded content quickly, Illustroke, which allows users to create vector images from text prompts, Patterned, which generates exact patterns for design needs, and Stockimg, which generates the perfect stock photo every time.

In conclusion, the application of AI-based services in the computer vision domain has significantly advanced the way we approach image and video generation. With their ability to generate high-quality and semantically meaningful images and videos, these AI-based services have a wide range of applications and are set to continue transforming the field of computer vision in the future.

B. Composing Music

AI-based services for Music Generation have created new opportunities for music creation and distribution. $Amper$ is an AI-powered music composition tool that uses machine learning algorithms to generate professional music based on the user input. Artificial Intelligence Virtual Artist (AIVA) is a computer program that can analyze a large dataset of music and use the information it learns to compose new pieces of music. MuseNet, developed by OpenAI, is a deep neural network that can generate 4-minute musical compositions with various instruments and styles. Soundraw is another service that allows users to create unique, royalty-free music. Jukedeck is a UK-based AI music composition platform that uses machine learning algorithms to generate original, royalty-free music. The platform allows users to specify the mood, length, and style of the music they want, and then uses artificial intelligence to generate a unique piece that meets their requirements. Jukedeck utilizes a deep neural network to analyze music data and generate new compositions. DeepSinger, developed by Baidu, is an AI-powered singing synthesizer that can generate songs in any language. The technology behind DeepSinger is GANs. The model is trained on a large dataset of singing voices and then uses that knowledge to generate new singing performances in real-time. This technology enables DeepSinger to produce human-like singing voices that are unique and realistic. These services provide an accessible and convenient way to generate music, making it possible for anyone, regardless of their background or experience in music, to create professional-quality compositions.

C. Content Generation Platforms

Generative AI for content creation became headline for last few months. The developers claim to generate content that is coherent, relevant, easier and faster to create new content.

OpenAI's large language model Generative Pre-Trained Transformer (GPT-3) has the ability to generate human-like text and can be used for various applications, including content creation. The model has been trained on a massive amount of data and can generate articles, stories, summaries, and even poetry. The model uses deep learning algorithms to analyze the input data and generate relevant and coherent outputs. Another AI-based content generation service is called $ArticleForge$. It uses advanced algorithms to analyze websites, keywords, and topics to generate unique articles. The service also has an intuitive interface that allows users to customize the generated content to meet their specific needs. According to the service provider, it can generate articles on various topics, including technology, health, business, and many more. Also there is tool $Wordsmith$ for content generation that can produce high-quality reports, summaries, and articles. The tool uses NLP algorithms to analyze data and produce content that is informative, engaging, and relevant. The platform can process large amounts of data and generate reports in minutes, making it ideal for businesses that need to produce regular reports.

$Zendesk$ offers a cloud-based customer service platform with features such as ticket management, analytics, reporting, and a chatbot using NLP and machine learning algorithms. $Helpshift$ is an AI-powered customer service platform with a chatbot and messaging features, offering ticket management, analytics, and reporting. Tars is an AI-powered customer service chatbot that utilizes NLP and machine learning algorithms to provide personalized and accurate responses. It can be integrated into various platforms and be trained on specific industries. Tars also offers features such as ticket management and analytics. HuggingFace's transformer library is a collection of pre-trained NLP models, including BERT, RoBERTa, and T5, that can be fine-tuned for specific use cases. BERT and RoBERTa are pre-trained models for NLP tasks such as text classification and question answering. T5 is pre-trained for NLP tasks such as text classification and text generation. AWS Comprehend is an NLP service from Amazon Web Services that enables businesses to extract insights from unstructured text data. It offers features such as entity recognition, sentiment analysis, and topic modeling. The service can be integrated with other AWS services for real-time data processing and analysis, allowing for quick decision making. Copy is an AI tool that generates copy with the aim of increasing conversions. Unbounce Smart Copy allows users to write high-performing cold emails at scale using AI. PuzzleLabs helps to build an AI-powered knowledge base for a user's team and customers workflows. Quickchat automates customer service chats, while Otter allows users to capture and share insights from their meetings.

D. Speech synthesis

AI-based speech synthesis algorithms are used to generate speech that mimics the voice of a specific individual [29]. One of the prominent examples of AI-based speech synthesis is the platform $LyrebirdAI$. It allows users to create custom speech and voiceovers using machine learning techniques. The platform models the speech patterns and characteristics of an individual by analyzing a sample of their speech. Once the voice is modeled/trained, the platform can generate new speech in that person's voice. This technology can be used for a variety of applications such as creating voiceovers for videos, podcasts, and commercials, as well as for speech-based interfaces and virtual assistants. $AdobeVoco$ is another example of AI-based speech synthesis technology that uses deep learning algorithms to generate new dialogue for a video based on a sample of speech from the original video. The generated speech sounds as natural as possible, making the technology useful for video editing. Krisp, Podcastle, and Descript are other examples of AI-based speech synthesis tools that use deep learning algorithms to provide advanced audio and video editing capabilities. $Krisp$ tool can remove background noises, echo, and other distractions from user's calls to provide high-quality audio. $Podcastle$ allows users to record studio-quality audio right from their computer, while Descript is a full-featured audio and video editing software powered by AI.

V. Summary

This review provides a brief description of Generative AI technologies and shows an historical perspective of their evolution. Earlier works mostly used function optimization of fitness (in attribute/feature space), and needed deeper knowledge of the algorithm to be used, also the computational power was very limited for large-scale data and applications. Over the years, exponential progress happened in both computing hardware (multi-core to many-core special processors) and software platforms; also natural language and image processing techniques improved significantly making the AI applications more efficient and user-friendly.

Most of the Generative AI techniques are reincarnation of these old efforts. For example, ChatGPT for code generation follows the steps of Genetic Programming introduced by John Koza of Stanford [30] and DALL-E2 for generating digital images/videos from natural language descriptions which follows the steps of Genetic Arts (by Karl Sims of MIT at [2]).

The current Generative AI works mostly use DNN (loss function minimization) for component (imagelet, nodes or codelets) ensembles where the output can be in any form as per the application. In these tools, additional layers are incorporated (to map text to codelet or text to imagelet) and some researchers are adding voice layer to make these more user-friendly (such as incorporating siri, alexa, etc.). The promoters of Generative AI tools argue that these will replace current Internet search engines by providing more specific context-aware information and tailored query-responses.

Generative AI technologies have shown remarkable success in variety of applications. For example, linguistic art and animation tools (WOMBO Dream, NightCafe, starryai, Midjourney, GPT, etc.) producing realistic images/videos, writing essay/script, codes, etc. However, most of these are relied on pre-trained models requiring extremely large data and significant computational power for training. So the limitations of these models exist in fine-tuning for specific tasks–avoiding bias, retaining long-term memory, interpretability, etc. Recent work "Memory Augmented Large Language Models are Computationally Universal" shows that an large language models (LLMs) can be enhanced with an associative read-write memory to extend ChatGPT like applications to any input size.

It is to be noted that the AI technology (like others) is not neutral, it is a double-edged sword and can be used for multiple purposes; while Generative AIs have been receiving both praise and criticism for their far reaching implications. In some applications, these can also create stories and video to spread disinformation to influence people and politicians. Adversaries may leverage Generative AI like ChatGPT (chatbot) to design zero-day attacks, develop sophisticated malware (such as ransomware, APTs, etc.), since these open-source tools can lower the bar for launching more advanced criminal activities. However, it may be possible to find markers/signatures to identify machine generated documents/objects since these are

[2] httpswww.karlsims.com/

Domain	Usage	Algorithm*	Product/Service
Computer Vision	Generate images from text descriptions.	Generative AI	Illustroke
			Wombo Team
			Flair
			Midjouney
	Generating videos with 3D models and animations.	VQVAE based model	Autodessys
			Wibbitz
			Synthesia
	Automated video-analysis and generation		DeepVideo
		Altair and Orion	Starryai
	Generating art from text.	VQGAN+CLIP	NightCafe
Music Generation	Composes personalized music based one emotions.	GAN	Jukedeck
		GAN+collaborative filtering	Amper Music
			AIVA
			Melodrive
		GAN+collaborative filtering+stylometry	MUSEnet
	Multi-lingual and Multi-singer singing voice synthesis.	lyrics-to-singingalignment	DeepSinger
Content Generation	Original long form articles and content generation from prompts.	Bert + GAN	Helpshift
			Article Forge
			Wordsmith
			Otter
			Helpshift
			Puzzle
			Zendesk
			Tars
			Copy
			Quickchat
			GPT-3
	Text analysis and sentiment analysis	Stylometry+DNN+NLP	AWS Comprehend
	Text generation and completion	BERT, Robert, T5, XLnet, CTRL, ALBERT	OpenAI's GPT-2
			HuggingFace
Speech Synthesis	Voice imitation and generating human-like speech from text.	GAN-TTS	Lyrebird AI
			Descript
			Adobe Voco
			Krisp
			Podcastle

TABLE I: A summary of generative AI products and services.(*information about algorithms are collected from different blog/media publication not from official source)

produced from a library of elements using a pattern-driven methods.

Researchers need to develop secure and trustworthy Generative AI applications. Adversarial Machine Learning (AML) studies demonstrated that it is possible to (mis)guide/classify deep learning systems by manipulating input data or modifying the latent space of pre-trained model parameters. We suggest that proper defense strategies such as NIST AI Risk Management Framework [31], Dual-Filtering techniques [32] should be integrated to develop trustworthy Generative AI applications.

Also for continued progress of generative AIs and make bigger impacts in society, more interesting and challenging problems need to be handled. For example, if the environment is unpredictably dynamic, uncertain, misleading, and have man-made obfuscation where behavior profiling or knowledge patterns are difficult to harness, how these techniques will perform with maximum accuracy in an uncontrolled environment. The current generative AI successes are possible not only because of algorithmic evolution but also increased computational capabilities i.e. exponential growth in hardware (computational power, storage capacity), cloud computing and related operational software.

REFERENCES

[1] D. Dasgupta, "Ai vs ai: Viewpoints," Tech. Rep. CS-19-001, The University of Memphis, May 2019.

[2] C. Tsai, C. Lai, H. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *J. Big Data*, vol. 2, p. 21, 2015.

[3] D. Dasgupta and Z. Michalewicz, "Evolutionary algorithms—an overview," *Evolutionary algorithms in engineering applications*, pp. 3–28, 1997.

[4] K. Sims, "Genetic arts." GenArts, Inc.

[5] C. Reynolds, "Evolutionary computation and its application to art and design."

[6] J. de Villiers, K. Hobbs, and B. Hollebrandse, “Recursive complements and propositional attitudes,” *Recursion: Complexity in cognition*, pp. 221–242, 2014.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[8] S. Schwanauer and D. Levitt, *AI and Music: Generative Grammars*. MIT Press, 1993.

[9] R. Agrawal, R. Srikant, *et al.*, “Fast algorithms for mining association rules,” in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, vol. 1215, pp. 487–499, Citeseer, 1994.

[10] C. Borgelt, “An implementation of the fp-growth algorithm,” in *Proceedings of the 1st International Workshop on Open Source Data Mining (OSDM)*, August 2005.

[11] “Ai illustrator: Art illustration generation based on generative adversarial network,” *IEEE*.

[12] M. M. Suh, E. Youngblom, M. Terry, and C. J. Cai, “Ai as social glue: Uncovering the roles of deep generative ai during social music composition,” 2021.

[13] X. Tu, Y. Zou, J. Zhao, W. Ai, J. Dong, Y. Yao, Z. Wang, G. Guo, Z. Li, W. Liu, and J. Feng, “Image-to-video generation via 3d facial dynamics,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 1805–1819, 2022.

[14] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018.

[15] J. Sun, Q. V. Liao, M. Muller, M. Agarwal, S. Houde, K. Talamadupula, and J. D. Weisz, “Investigating explainability of generative ai for code through scenario-based design,” 2022.

[16] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations, ICLR* (Y. Bengio and Y. LeCun, eds.), 2014.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in *Annual Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, pp. 5998–6008, 2017.

[19] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), pp. 4171–4186, Association for Computational Linguistics, 2019.

[20] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners,” 2019.

[21] 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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[22] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli, “Understanding the capabilities, limitations, and societal impact of large language models,” *CoRR*, vol. abs/2102.02503, 2021.

[23] Y. Liu, J. Zhang, H. Xiong, L. Zhou, Z. He, H. Wu, H. Wang, and C. Zong, “Synchronous speech recognition and speech-to-text translation with interactive decoding,” in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pp. 8417–8424, AAAI Press, 2020.

[24] F. Weisser, T. Mayer, B. Baccouche, and W. Utschick, “Generative-AI Methods for Channel Impulse Response Generation,” in *25th International ITG Workshop on Smart Antennas (WSA 2021), French Riviera, France*, pp. 1–6, 2021.

[25] E. Dervakos, G. Filandrianos, and G. Stamou, “Heuristics for Evaluation of AI Generated Music,” in *25th International Conference on Pattern Recognition (ICPR), Milan, Italy*, pp. 9164–9171, 2021.

[26] A. Razavi, A. Van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with vq-vae-2,” *Advances in neural information processing systems*, vol. 32, 2019.

[27] K. Crowson, S. Biderman, D. Kornis, D. Stander, E. Hallahan, L. Castricato, and E. Raff, “Vqgan-clip: Open domain image generation and editing with natural language guidance,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, pp. 88–105, Springer, 2022.

[28] C. Lei, S. Luo, Y. Liu, W. He, J. Wang, G. Wang, H. Tang, C. Miao, and H. Li, “Understanding chinese video and language via contrastive multimodal pre-training,” in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2567–2576, 2021.

[29] X. Xu *et al.*, “VSEGAN: Visual Speech Enhancement Generative Adversarial Network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore*, pp. 7308–7311, 2022.

[30] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT press, 1992.

[31] N. I. of Standards and Technology, “Ai risk management framework,” 2022. [Online; accessed 4-February-2023].

[32] D. Dasgupta and K. D. Gupta, “Dual-filtering (df) schemes for learning systems to prevent adversarial attacks,” *Complex & Intelligent Systems*, pp. 1–22, 2022.

[33] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, “Deepsinger: Singing voice synthesis with data mined from the web,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1979–1989, 2020.

Appendix

Web URL for the app services list

- Illustroke https://www.Illustroke.com/
- Wombo Team: https://www.wombo.ai/
- Flair: https://flair.co/
- Midjouney https://www.Midjouney.com/
- Autodessys: https://www.autodessys.com/
- Wibbitz: https://wibbitz.com/
- Synthesia https://Synthesia.io/
- DeepVideo: https://deepvideo.com/
- Starryai: https://starry.ai/
- NightCafe https://www.NightCafe.studio/
- Jukedeck: https://www.jukedeck.com/
- Amper Music: https://ampermusic.com/
- AIVA: https://www.aiva.ai/
- Melodrive: https://www.melodrive.com/
- MUSEnet: https://openai.com/research/musenet/
- DeepSinger [33]
- Helpshift: https://www.helpshift.com/
- Article Forge: https://articleforge.com/
- Wordsmith: https://automatedinsights.com/wordsmith/
- Otter: https://otter.ai/
- PuzzleLabs https://PuzzleLabs.ai/
- Zendesk: https://www.zendesk.com/
- Tars: https://www.tars.com/
- Copy: https://copy.ai/
- Quickchat https://Quickchat.ai/
- GPT-3: https://openai.com/research/gpt-3/
- AWS Comprehend: https://aws.amazon.com/comprehend/
- OpenAI’s GPT-2: https://openai.com/research/gpt-2/
- HuggingFace: https://huggingface.co/
- Lyrebird AI: https://lyrebird.ai/
- Descript: https://www.descript.com/
- Krisp: https://krisp.ai/