JEODEZİ VE JEOİNFORMASYON DERGİSİ
JOURNAL OF GEODESY AND GEOINFORMATION

ISSN : 2147-1339
e-ISSN: 2667-8519

Araştırma Makalesi / *Research Article*
Yıl / *Year:* 2024 **Cilt /** *Volume:* 11 **Sayı /** *Issue*: 2
Sayfa / *Page*: 116-148
Dergi No / *Journal No*: 120
Doi: 10.9733/JGG.2024R0009.T

Üretken uzamsal zekânın getirdiği paradigma değişimi

Caner Güney[1*]

[1]*İstanbul Teknik Üniversitesi, Maslak Kampüsü, İnşaat Fakültesi, Geomatik Mühendisliği Bölümü, Sarıyer, İstanbul.*

Öz: *İnsanoğlunun düşünce sistemini kolaylaştıran ve tamamlayan, akıl yürütme ve çıkarım yapmasına olanak veren üretken yapay zekâ uygulamalarının önemli yükselişi, farklı endüstrilerden son kullanıcılara kadar hemen hemen her kesimin günlük yaşamını ve iş yapış şeklini değiştirmektedir. Üretken yapay zekâ; kişilerin ve kuruluşların verimliliklerini ve üretkenliklerini arttırmakta, yenilikçi ürün ve hizmetlerin geliştirilmelerini kolaylaştırmakta, zaman ve kaynak tasarrufu sağlamakta ve kısacası geleceğe yön vermektedir. Büyük dil modellerinin ve bununla birlikte üretken yapay öğrenme modellerinin birçok üstün yönü olmasına rağmen bazı kısıtlı kaldığı geliştirilmesi gereken yönleri de bulunmaktadır. Genel bilgiye sahip büyük dil modellerinin belirli alanlarda daha etkin kullanılabilmesi için öncelikle ince ayarlarının yapılması ve eksik arka plan bilgisinin kapatılması için bilgi çizgeleri ile bütünleştirilmesi gerekmektedir. Sonrasında ise JSON gibi standart web formatları ile sorunsuz çalışabilmesi sağlanmalı ve eylem modelleri ile ilişkilendirilmelerindeki yetersizlikler giderilmelidir. Çalışma kapsamında sözü edilen bu eksiklikler tartışılacak, kurum ve kuruluşların bu eksiklileri nasıl bir bilgi işlem altyapısı ile ele almaları gerektiği üzerinde durulacaktır. Üretken yapay zekâ uygulamalarından elde edilen deneyimin uzamsal/mekânsal zekâ ile bütünleştirilerek üretken uzamsal zekâ kavramının nasıl şekilleneceğine ilişkin çıkarımlar çalışma kapsamında paylaşılacaktır.*

Anahtar Sözcükler: Yapay zekâ, Uzamsal zekâ, Üretken yapay zekâ, Çok modlu yapay zekâ, Büyük dil modelleri

Paradigm shift by generative spatial intelligence

Abstract: *Generative AI (GenAI) applications facilitate and complement human thinking, enabling reasoning and inference. The significant rise of GenAI applications is changing the daily lives and the way of doing business for almost everyone, from different industries to end users. GenAI increases the efficiency and productivity of individuals and organizations, facilitates the development of innovative products and services, saves time and resources. In summary, it shapes the future. While there are many advantages of large language models and GenAI models, there are also some limitations. There are some issues where large language models with general knowledge are inadequate, such as fine-tuning them to be used more effectively in certain areas, integrating them with knowledge graphs to cover missing background information, working smoothly with standard web formats such as JSON, and associating them with large action models. Within the scope of the study, these deficiencies will be discussed. It will also focus on how organizations should configure an information processing infrastructure to overcome these deficiencies. The study will discuss how the concept of generative spatial intelligence (GenGeoAI) will be shaped by integrating the experience gained from GenAI applications with spatial intelligence (GeoAI).*

Keywords: Artificial intelligence, Spatial intelligence, Generative AI, Multi-modal AI, Large language models

* Sorumlu Yazar/Corresponding Author: Tel: +90 212 285 3825

0000-0002-1620-1347, guneycan@itu.edu.tr (Güney C.)*

Geliş Tarihi/Received: 01.04.2024
Kabul Tarihi/Accepted: 06.05.2024

1. Giriş

4. Endüstri Devrimi ile ortaya çıkan üretim şekillerinin değişim ve dönüşüm hızı COVID-19 pandemisiyle birlikte ivme kazanmıştır. 4. Endüstri Devrimi ile hızla ilerleyen dijital dönüşüm bugün kendini yapay zekâ dönüşümü olarak göstermektedir. Sözü edilen dönüşüm kural bazlı modeller, geleneksel analitikler yerine kendi kendine öğrenen ve geleneksel veri analitiğinin çok ötesinde veriden değer ortaya çıkaran yapay öğrenme algoritmalarının kullanılması biçiminde görülmeye başlanmıştır. ChatGPT, Q*, Gemini, Yapay Genel Zekâ (Artificial General Intelligence, AGI), Figure AI örneğindeki gibi yapay zekânın robotlarla buluşması, 5. Endüstri Devrimi vb. yaklaşımlar yapay zekâ dönüşümünün önemli çıktıları olarak görülebilir. Yapay zekâ dönüşümüyle birlikte bugüne kadar insan ile yapay zekâ etkileşimi hiç olmadığı kadar yüksek düzeyde gerçekleşmektedir. Ayrıca bu etkileşim giderek artmakta ve etkileşimli öğrenme biçiminde ilerlemektedir.

İnsanlık bugüne kadar yapay öğrenmeden (yapay zekâdan, makine öğrenmesinden) bir hesaplama aracı olarak yararlanmıştır. Son zamanlarda yapay öğrenme yalnız bir hesaplama aracı olarak değil aynı zamanda bir üretim aracı olarak kullanılmaya başlanmıştır. Bu noktadan sonra ise daha yüksek hesaplama güçleri, daha çok veri ve çok modlu yapay öğrenme modelleri ile insanoğlunun problem çözme kapasitesini ve karar verme yeteneğini arttıracak daha farklı uygulamaların ortaya çıkması beklenmektedir. Bu nedenle bu çalışma yapay zekâ dönüşümünün mekânsal bilişim alanına yansıması konusuna değinmektedir.

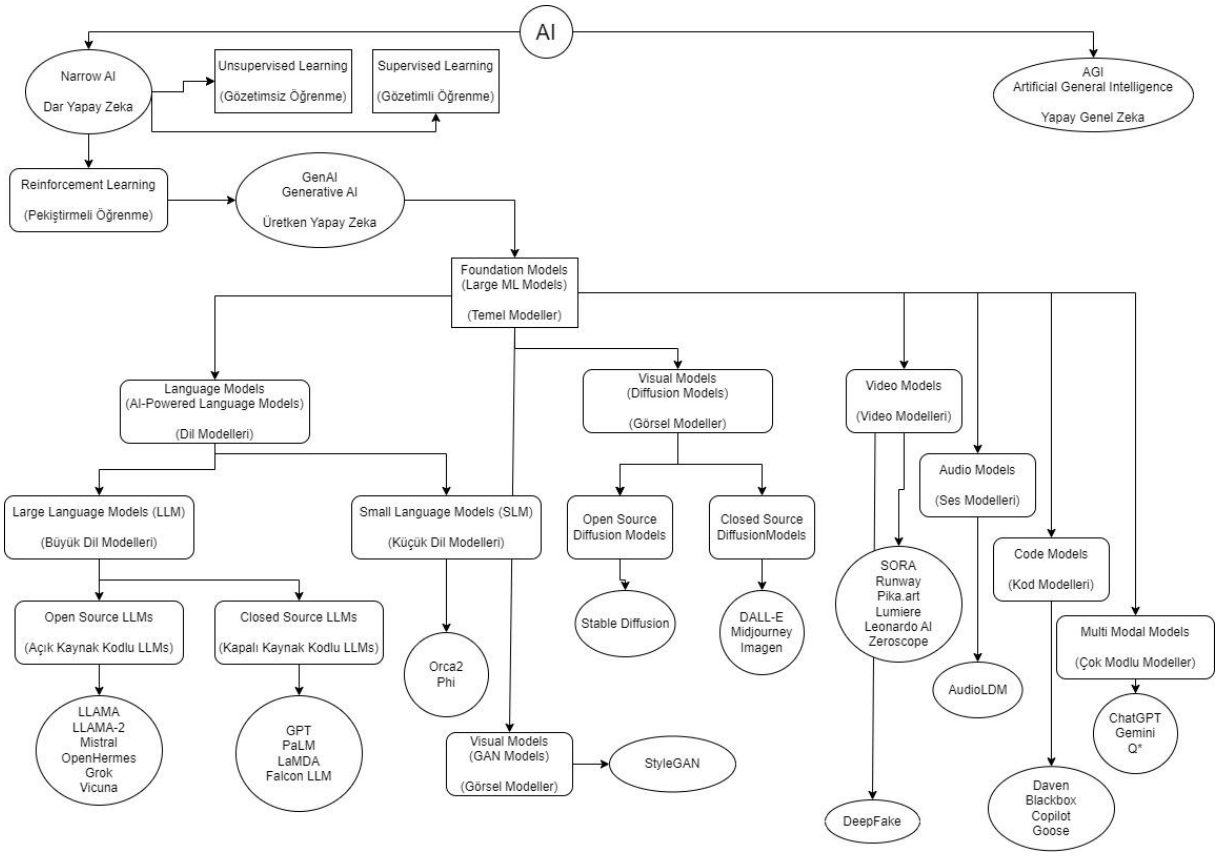
Bununla birlikte yapay zekâ yalnız bilim, teknoloji, iş geliştirme, ekonomi açılarından değil etik ve varoluşsal konuları da içeren felsefe, sosyoloji, antropoloji, psikoloji bakış açılarından da ele alınması gereken bir kavramdır. Yapay öğrenme kavramı üzerinde farklı açılardan düşünülmediğinde ön yargıları olan uygulamalar ve ön yargı ile karar verebilen bilgi sistemleri ortaya çıkabilmektedir.

Biyolojik temelli zekâ olan doğal zekâyı tam olarak tanımlayamayan, düşünmenin nasıl gerçekleştiğini tam olarak modelleyemeyen insanoğlu; insan gibi düşünen, karar veren ve eyleme geçen makineleri, sistemleri, robotları ortaya çıkarmak için yoğun biçimde çalışmaktadır. Hatta bununla da sınırlı kalmayıp doğal zekâyla, silikon temelli yapay zekâyı neuralink vb. projeler üzerinden bütünleştirmeye çalışan teknolojileri hayata geçirmeye başlamıştır.

Üretken Yapay Zekâ (Generative AI, GenAI), mevcut yapılar hakkındaki veri kümelerinden öğrenen ve bunu yeni yapılar oluşturmak için kullanan yapay öğrenme tekniklerine denir. Bunun için çok büyük etiketsiz veri kümeleri üzerinde eğitilmiş büyük makine öğrenim modelleri olan ve Şekil 1'de gösterilen temel modeller (*foundation models*) kullanılmaktadır. Temel modeller arasında da insanın düşünce biçimini yansıtan büyük dil modelleri öne çıkmaktadır. Büyük dil modellerini kullanan ChatGPT gibi üretken uygulamalar makine öğrenimi kullanımının tabana yayılması şeklinde görülmektedir. Öyle ki ChatGPT uygulamasının kullanıcı sayısı 5 günde 1 milyona ulaşmış ve 2 ayda 100 milyonu geçmiştir. Üretken yapay öğrenme yalnız metin üreten bir teknoloji olmayıp, ses, görüntü, video, yazılım kodu gibi içerikler de oluşturabilmektedir. Bu açıdan bakıldığında yukarıda da ifade edildiği gibi insan-makine, insan-yapay zekâ etkileşiminin hiç bu kadar yüksek düzeyde olmadığı bir dönem yaşanılmaya başlanmıştır. Tüm dünyada çok fazla insanın üretken yapay zekâ uygulamalarını kullanması, makine öğreniminin demokratikleşmesi olarak yorumlanmaktadır.

Bugüne kadar yapay öğrenme alanında ortaya çıkan gelişmeler insanoğlunun çeşitli iş alanlarına olan yaklaşımını değiştirmekte ve yeniden şekillendirmektedir. Bununla birlikte teknoloji ve insan etkileşiminin geleceğinin şekillenmesi Google, Microsoft, Meta, Apple, OpenAI, Nvidia gibi büyük işletmelerin öncülüğünde diğer bir ifadeyle küresel sermayenin liderliğinde gerçekleşmektedir. Bu işletmelerin dışında kalan geliştiricilerin üretimi çoğunlukla belirli alanlar için özel modellerin ince ayarının yapılması ile sınırlı kalmaktadır. Bu durumda yapay öğrenmenin demokratikleşmesi yaklaşımı

aslında arkasında büyük işletmelerce geliştirilen temel modellere ve hemen arkasından bu modellere dayalı geliştirdikleri uygulamalara bağımlılıkları bulundurmaktadır. Sözü edilen büyük işletmeler sahip oldukları büyük miktardaki veriden yine sahip oldukları büyük miktardaki hesaplama gücünü kullanarak temel modelleri oluşturmaktadır. Bu durumda veriyi nasıl elde ettikleri, veri kalitesinin ne düzeyde olduğu, hangi tür veri kümelerine daha fazla ağırlık verildiği vb. konular ön yargılı uygulamaların ve sistemlerin ortaya çıkmaması için önemli bir hâl almaktadır.



Şekil 1: *Yapay zekâ kavramında temel modeller*

Makine öğrenmesinin demokratikleşmesi diğer bir ifadeyle tabana yayılması konusuna ilişkin üretken yapay zekâ yolculuğunda sahip olunan yeteneklere ve kaynaklara bağlı olarak aşağıdaki seçenekler ortaya çıkmaktadır:

- **Önceden eğitilmiş (*pre-trained*) modellerin kullanımı:** Başkasının verisi ile başkasının modeli kullanılarak eğitilmiş modelin doğrudan kullanımıdır. Bu tür modellerin kullanımı kolaydır, makine öğrenmesi konusunda bilgi sahibi olmayı gerektirmez. Örneğin son kullanıcılar hiç kodlama yapmadan metin, görüntü, video, ses gibi içerikleri üretken yapay zekâ uygulamaları ile üretebilmektedir.
- **Yeniden eğitilmiş (*re-trained*) modellerin kullanımı:** Başkasının kullanıma hazır modelinin kullanım amacına uygun olan veri kümesi ile tekrardan eğitilmesidir. Bu yaklaşım ince ayar (*fine-tuning*) olarak da bilinir ve gerçekleştirmek için yapay öğrenme konusunda belirli düzeyde bilgi sahibi olmak gerekir.
- **Özel (*custom*) modellerin kullanımı:** Kişi veya kuruluşların kendisine ait veri kümesiyle kendi modelini geliştirerek kullanmasıdır. Bu tür modelleri geliştirmek (*full training*) için yapay öğrenme konusunda ileri düzeyde bilgi sahibi olmak ve büyük bilgi işlem altyapısına sahip olmak gerekir.

Bu durumda öncelikli yapılabilecekler arasında üretken yapay öğrenme modellerinin üretim ve hizmet sektörlerinde, mühendislik uygulamalarında etkin kullanımı için kullanım senaryolarının oluşturulması yer almaktadır. Birçok sektör için iş yapış şekilleri yeni bir boyuta taşınacak, otomatik öneri sunan sistemler/uygulamalar yaygınlaşacak, kişisel asistanların

kullanımı artacaktır. Diğer sektörlerde olduğu gibi Geomatik/Harita mühendislerinin, mekânsal bilişim alanında çalışan uzmanların yaptıkları işler için kullandıkları mevcut üretim bantları da değişecektir. Hatta mekânsal zekâ uygulamaları (Güney, 2019) üretken mekânsal zekâ uygulamalarına dönüşecektir. Bir başka ifadeyle mekânsal bilişim alanında kullanılan yapay öğrenme modelleri yerini üretken yapay zekâ öğrenme modellerine bırakacağı düşünülmektedir. Bu durum çalışmanın odak noktasını oluşturmaktadır.

Büyük dil modellerinin dolayısıyla üretken yapay zekânın güçlü yönleri olduğu gibi insan-makine etkileşiminin daha verimli olabilmesi için geliştirilmesi gereken zayıf ya da eksik kalmış yönleri veya sınırlamaları da bulunmaktadır. Bu durumu değiştirmek için aşağıdaki konular üzerine çalışmaların yapılması kişi, topluluk ve kuruluşların gerçekleştirebileceği konular arasında yer alabilir:

- İnce ayar ile büyük dil modellerinin bilgi yoğun istemlere yanıt verme doğruluğunun artırılabilmesi,
- Büyük dil modellerinde yapılandırılmış bilgiyle mantıksal akıl yürütebilmenin sağlanabilmesi,
- Büyük dil modelleriyle doğal dil sorgularından yapılandırılmış bilginin üretilebilmesi,
- Büyük dil modellerinde yapılandırılmış veri kümeleriyle birlikte analitik ve analiz yapılabilmesi için uygun hesaplama mimarilerinin geliştirilebilmesi,
- Büyük dil modellerinde ve üretken yapay zekâ uygulamalarında planlama algoritmalarının kullanılabilmesi,
- Büyük eylem modelleri ile büyük dil modellerinin yapabilirliklerinin artırılması.

Çalışma kapsamında yukarıda maddeler halinde ifade edilen konulara bağlı olarak üretken yapay öğrenme modellerinin daha etkin kullanılabilmesi için yapılması gerekenler tartışılmaya çalışılmıştır. Bu tartışmalara bağlı olarak üretken uzamsal zekâ modelleri geliştirme üzerine bazı çıkarımlar yapılmıştır.

2. Büyük Dil Modelleri

İnsanlar tarafından kullanılan doğal dil ve makineler tarafından kullanılan makine dili olmak üzere genel olarak birbirinden tamamen farklı iki tür dil bulunmaktadır. Buna ek olarak, yapay öğrenmeyle (makine öğrenmesiyle) birlikte insan ile makine arasında iki yönlü bir etkileşim oluşmaya başlamıştır. Makinelerin insan dilini anlaması ve yorumlaması "Doğal Dil İşleme (Natural Language Processing, NLP)" alanının konusudur. İnsan tarafından makineye verilen metinsel ifadeyi analiz eden ve metinden herhangi bir "şeye" geçiş sağlayan (text-to-anything, text-to-X) ChatGPT vb. uygulamalar doğal dil işleme yöntemini kullanmaktadır. Doğal dil işleme teknikleri daha önce 2010'lu yıllarda Siri (Apple – 2011), Cortona (Microsoft – 2014), Alexa (Amazon – 2014), Assistant (Google – 2016), Bixby (Samsung – 2017) gibi akıllı asistanlarda ve hemen arkasından Google Translate ve DeepL Translator gibi çözümler üzerinden farklı diller arasında çeviri gerçekleştirmede etkin olarak kullanılmıştır. Bugün ise ChatGPT (OpenAI – 2022), Gemini (Google – 2023), Copilot (Microsoft – 2023) vb. üretken yapay zekâ uygulamaları "Büyük Dil Modellerini (Large Language Models, LLMs)" kullanarak insanın; makineyle, bilgisayarla, yapay zekâyla bugüne kadar hiç olmadığı kadar yüksek düzeyde etkileşim kurabilmesini sağlamaktadır.

Bir tümcedeki sözcüklerin sırasının, bu sözcüklerin anlamlı olma olasılığına göre belirlenmesi dil modellemenin (*language modeling*) konusudur. Büyük dil modelleri insan dilini işleme (*language processing*), insan gibi metin oluşturma (*generation*) ve dil uyarlama (*adaptation*) için güçlü araçlardır. Büyük dil modelleri; girdiyi anlamak için "Doğal Dil Anlama (Natural Language Understanding, NLU)" ve bir yanıt formüle etmek için "Doğal Dil Oluşturma (Natural Language Generation, NLG)" yöntemlerini kullanmaktadır. Büyük dil modelleri çevrimiçi kitaplar, Wikipedia gibi insan bilgisinin ve iletişiminin geniş bir yelpazesini kapsayan farklı ve kapsamlı metin derlemeleri (*text corpuses*) üzerinde eğitilmektedir. Büyük dil modelleri eğitildiği kaynaklarda bulunan bilgiler arasındaki ilişkileri kurar ve örüntüleri (*pattern*) yakalar. Bu eğitim

sonucunda metinsel ifadeler üzerinden dildeki sözcüklerin dağılımına, yani dile ilişkin temel istatiksel bilgiye sahip olurlar. Sahip oldukları bu bilgi, sözcüklerin tahmin edilmesinde kullanılmaktadır. Bu durumda model; söz dizimi (*syntax*), dil bilgisi (*grammar*), anlam bilim (*semantics*) veya üst düzey bilim öğrenmek arasında bir fark görmemekte ve yalnız bir sonraki sözcüğü tahmin etmede (*predict*) daha iyi olmak için uğraşmaktadır. Böylece, büyük dil modelleri göreve özel (*task-specific*) kapsamlı bir eğitime gereksinim duymadan dille ilgili çeşitli görevleri farklı alanlarda, konularda ve yazma şekillerinde etkili bir şekilde yerine getirebilmektedir.

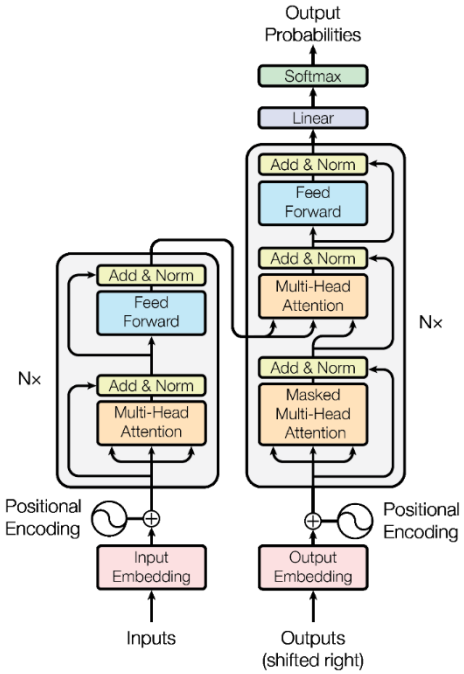
Büyük dil modelleri genel olarak insan dilini anlamak ve insan gibi metin üretmek için büyük miktarda metin veri kümeleri üzerinde önceden eğitilmiş (*pre-trained*) yapay öğrenme modelleridir. Büyük dil modeli, eğitiminde kullanılan tüm metinsel ifadelerin bir istatistiğini oluşturur ve bu istatistiksel söz dizimine dayalı parametreleri temel alarak kullanıcının uygulama üzerinden girdiği istemlere (*prompt*) tamamlayıcı olacak en olası bir dizi sözcüğü tahmin eden bir fonksiyon olarak düşünülebilir. Derin öğrenmeye dayalı dönüştürücü modellerini (*language transformers*) kullanan büyük dil modelleri sıralı/dizili verinin (*sequential data*) yorumlanması ve yönetilmesi konusunda gelişmiştir. Bu sayede bağlama duyarlı (*contextually-aware*) konuşmaları anlayabilmekte ve bu bağlamla ilgili tutarlı yanıtlar üreterek konuşmayı sürdürebilmektedir. Büyük dil modelleri, metinsel ifadelere ilişkin olasılık dağılımlarını, sözcüklerin birlikte bulunma olasılıklarını ve metinsel ifadeye ilişkin istatistik bilgileri kullanarak tümevarım çıkarımı (*inductive inference*) da gerçekleştirebilmektedir.

Büyük dil modelleri anlamsal ve söz dizimsel tümce yapılarını yorumlayarak ve analiz ederek çok yönlü dil işleme görevlerini etkin bir şekilde gerçekleştirebilir. Öyle ki, büyük dil modelleri; metin oluşturma, metin tamamlama, çeviri, özetleme, yeniden yazma, soru yanıtlama, metin sınıflandırma gibi çeşitli doğal dil işleme görevlerini başarıyla yerine getirme yetenekleri nedeniyle son zamanlarda büyük ilgi ve popülerlik kazanmışlardır. Bununla birlikte, büyük dil modeli temelli üretken yapay zekâ (Generative AI, GenAI) uygulamalarında halüsinasyon, yanlılık gibi teknik problemler bulunmaktadır. Söz edilen bu problemler; üretken yapay zekâ uygulamalarının ilk bakışta makul gelen ancak gerçekte yanlış bilgileri savunan metinler oluşturma, eksik veya yanlış veri kümelerine dayalı varsayımlarda bulunma, anlamsız döngülere girme, akademik literatür uydurma vb. sorunları ortaya çıkarmaktadır. Büyük dil modelleri eğitildikleri veri kümelerinin yanlılıklarına ve sınırlamalarına sahip oldukları için yanlı ve yanlış çıktılar verebilmektedir. Büyük dil modelleri muğlak ve çelişkili girdilerle karşılaştıklarında belirsiz ve tutarsız yanıtlar üretebilmektedir. Bu durum büyük dil modellerinin başarımını ve üretken yapay zekâ uygulamalarının güvenilirliğini sorgulatmaktadır.

Üretken yapay öğrenme uygulamaları ya da büyük dil modelleri her ne kadar kullanışlı olsa da geliştirme maliyetlerinin yüksekliği, ayrıntı düzeyi sorunu, akıl yürütme için ince ayar gerektirmesi gibi zorlukları faydalarının önüne geçebilmektedir.

2.1 Büyük Dil Modellerinin Çalışma Şekli

Büyük dil modelleri dönüştürücü tabanlı modellerdir (*transformer models*). Dönüştürücü bir veri dizisinin tüm parçaları arasındaki ilişkiyi bütünsel olarak öğrenen bir tür yapay sinir ağıdır. Dönüştürücüler Şekil 2'de görüldüğü üzere kodlayıcı (*encoder*) ve kod çözücü (*decoder*) olmak üzere temelde iki ana bölümden (*module*) oluşmaktadır. Dönüştürücülerin kullanımında birden fazla kodlayıcı ve kod çözücü yapısı kullanılmaktadır (Vaswani vd., 2023). Vaswani vd. (2023) çalışmalarında bu değeri (Nx) 6 olarak belirlemiştir.



Şekil 2: *Dönüştürücü mimarisi (Vaswani vd., 2023)*

Kodlayıcı bölümün girdisi metinsel ifadedir. Ancak yapay öğrenme modelleri metinsel temsilleri (*text representation*) doğrudan kullanamamaktadır. Kodlayıcı bölümde sıralı metin dizisi işlenir ve sayısal temsile (*numerical representation*) dönüştürülür. Bir diğer ifadeyle metinsel ifade bağlamsal ilişkileri çıkarabilmek için kodlanır. Sözcüklerin sayısal temsiline katıştırma (*embedding*) denir. Katıştırmalar metnin bağlamını (*context*) ve anlamını (*meaning*) yakalamak için kullanılır.

Kod çözücü bölüm girdi olarak kodlayıcı bölümde üretilmiş olan katıştırmaları alır ve analiz eder. Bir diğer ifadeyle kodlanmış metinlerin ilgili çıktıyı oluşturması için kodları çözülür. Kod çözücü bölüm çıktı olarak anlamlı, konuyla ilgili ve tutarlı insan benzeri metin dizileri üretir.

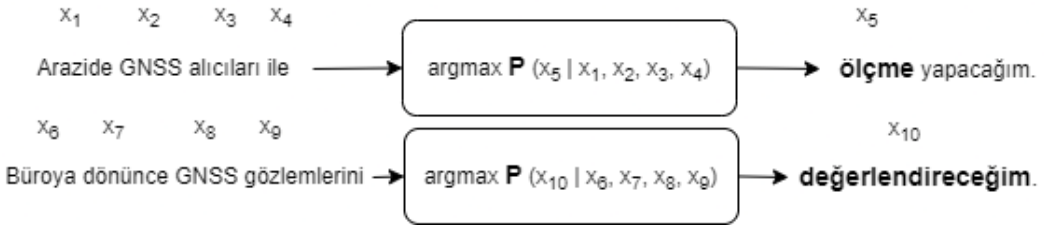
Büyük dil modellerindeki dönüştürücü modeller metinsel ifadenin önemli kısmını vurgulamak ve daha sonra bunu hatırlamak için dikkat düzeneğini (*self-attention mechanism*) kullanmaktadır (Vaswani vd., 2023). Bu yönüyle dönüştürücü modeller, kendinden önceki dikkat düzeneği kullanmayan yinelemeye (*recurrence*) dayalı olan RNN, RSTM, GRU gibi derin öğrenme tabanlı modellerden ayrılmaktadır.

Şekil 2'de görüldüğü üzere her kodlayıcı, metinsel ifadenin kendisine dikkat eden bir öz dikkat katmanına (*self-attention layer*) ve bir ileri beslemeli sinir ağı katmanına (*feed forward neural network layer*) sahiptir. Her kod çözücü de iki öz dikkat katmanına ve bir ileri beslemeli sinir ağı katmanına sahiptir. Alt katmanlar arasında yani öz dikkat katmanıyla ileri beslemeli sinir ağı katmanı arasında ekleme (*add*) ve normalleştirme (*normalization*) katmanları bulunmaktadır. Bunların görevi alt-katmandan gelen çıktıyı katman normalizasyonu tekniğiyle normalleştirmektir. Öz dikkat katmanına veya ileri beslemeli sinir ağı katmanına gitmeyen bazı bilgiler doğrudan normalleştirme katmanına gönderildiği Şekil 2'de görülmektedir. Buna atlama durumu (*skip state*) denilmektedir. Bunun amacı, modelin bazı şeyleri unutmamasını ve modelin önemli olan bilgileri ağın daha ileri aşamalarına iletebilmesini sağlamaktır. Şekil 2'de çok başlı dikkat (*multi-head attention*) ve maskeli çok başlı dikkat (*masked multi-head attention*) olmak üzere iki farklı dikkat katmanı görülmektedir. Genel olarak her ikisi de aynı işlevi görse de tek farkları çok başlı dikkat katmanında tüm sözcükler bir tümce içerisinde girilen diğer tüm sözcüklerle karşılaştırılırken, maskeli çok başlı dikkat katmanında yalnız bir sözcükten önce gelen sözcükler tümcedeki o sözcükle karşılaştırılmaktadır (Vaswani vd., 2023).

Hem kodlayıcının hem de kod çözücünün girdileri katıştırmalardır (*embeddings*). Dönüştürücü mimarisinde yinelemeler olmadığı için katıştırmalara konumsal kodlayıcı (*positional encodings*) eklenmektedir. Konumsal kodlayıcı sözcüğün tümce içerisinde nerede olduğunu diğer bir ifadeyle hangi sözcüğün önce ve diğerlerinin sonra geldiğini göstermek için kullanılır. Vaswani vd. (2023) çalışmalarında bunun için farklı frekanslarda sinüs ve kosinüs fonksiyonları kullanılmıştır. Konumsal kodlayıcılar ile iki sözcük arasındaki uzaklık hesaplanabilir (Vaswani vd., 2023).

Şekil 2'de görüldüğü üzere kod çözücü bölümünün sonunda çıktı için bir doğrusal katman (*linear transformation*) ve bir *softmax* katmanı bulunmaktadır. Burada vektördeki her bir sözcüğün, dizideki son kelimeden sonraki sözcük olma olasılıkları bulunmaktadır. Bu katmanlar modelin çıktılarını insanların anlayabileceği ifadelere dönüştürmektedir.

Büyük dil modelleri genel olarak şu şekilde çalışmaktadır. Modele girdi olan metinsel ifadeye önce alt sözcük birimi yapısına dönüştürme (*tokenization*) işlemi uygulanarak alt sözcük birimlerinin katıştırmaları (*token embeddings*) oluşturulur. Böylece her bir alt sözcük birimi (*token*) bir tanımlama sayısı (*token id*) almış ve metinsel ifadenin temsili bir matris formuna (*numerical representation*) dönüştürülmüş olur. Alt sözcük birimleri katıştırmalarına konumsal kodlayıcılar eklendikten sonra ortaya çıkan matris, ilk kodlayıcı bölümün çok başlı dikkat katmanına iletilir. Bu katmanda katıştırma matrisi sorgu (*query*), anahtar (*key*) ve değer (*value*) matrisleri ile çarpılır. Çarpım sonucunda her bir sözcük için sorgu, anahtar ve değer vektörleri elde edilir. Sorgu vektörü ile anahtar vektörünün skaler çarpımı (*dot product*) yapılarak her sözcük için tümcedeki diğer tüm sözcüklere karşılık gelen bir puan hesaplanır. Bir çeşit ağırlık (*attention weights*) olarak kabul edilebilecek bu puanlar normalleştirilerek *softmax* katmanına gönderilir. *Softmax* katmanında bu ağırlıklar sözcüklerin değer vektörleri ile çarpılır ve bağlam vektörü (*context vector*) elde edilir. Buna bağlı olarak *softmax* katmanı bir sonraki sözcüğün hangi sözcük olacağını belirler (Vaswani vd., 2023). Böylece Şekil 3'te görüldüğü gibi metin üretilebilir. Bu dizide beşinci sözcük, ilk dördü göz önüne alınarak en olası bir sonraki sözcük olarak tahmin edilmektedir.



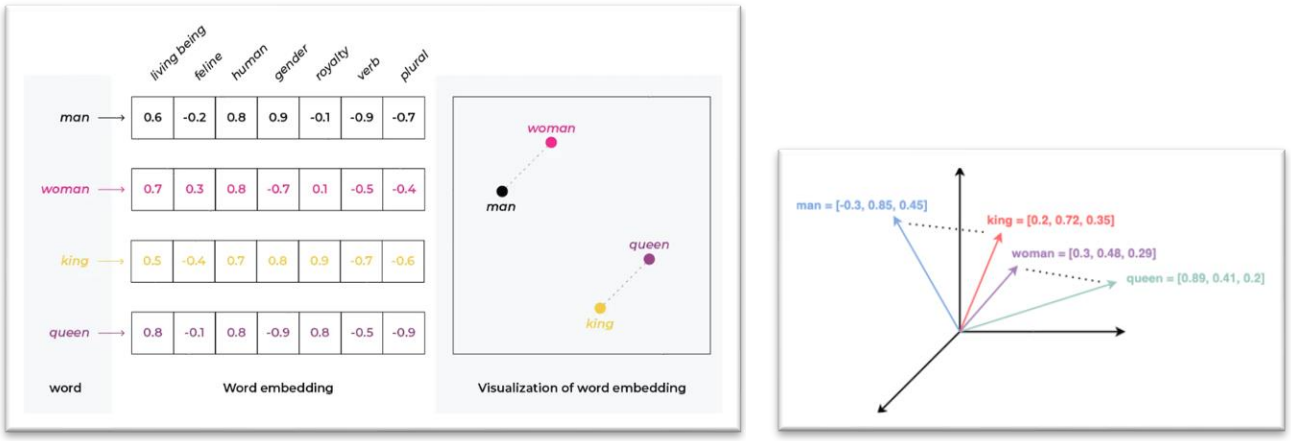
Şekil 3: *Katıştırmalara ve yapay sinir ağlarına dayalı tahmin görevi*

2.2 Katıştırma Uzayı

Doğal dil işlemede sözcük katıştırmaları (*word embeddings*) dışında metinsel ifadenin sayısal temsili konusunda sayım temelli (*count-based*), tek vuruşlu kodlama (*one-hot encoding*) gibi başka yaklaşımlar da bulunmaktadır. Sayım temelli yöntemde sözcüklerin tümce içerisinde kaç kez kullanıldığına (*occurance*) dayalı "bag of words", "N-gram", "TF-IDM" gibi bir dizi faklı yaklaşım bulunmaktadır. Doğal dil işleme alanında uzun zamandır kullanılan bu yöntemler; bağlamı dikkate almaması, eğitim sırasında karşılaşmadıkları sözcüklerle başa çıkamamaları, seyrek vektör oluşturmaları vb. nedenlerle verimli biçimde kullanılamamıştır. Bundan dolayı bugün doğal dil işleme alanında sözcük katıştırmaları yaklaşımı öne çıkmaktadır.

Sözcük katıştırma vektörleri katıştırma uzayında (*embedding space*) bulunmaktadır. Bu uzayda sözcük katıştırmaları arasındaki uzaklık ne kadar yakınsa, sözcükler de o kadar benzer veya aynı bağlamda kullanılan sözcüklerdir. Vaswani vd. (2023) çalışmalarında kullandığı örnek Şekil 4'te görüldüğü üzere kadın-erkek, kral-kraliçe sözcükleri benzer veya aynı bağlamda kullanılan sözcükler olduğundan aralarındaki uzaklıklar farklı bağlamlarda kullanılan diğer sözcüklere göre azdır.

Eğer erkek ve kadın sözcükleri katıştırma uzayında bu şekilde konumlanıyorsa, kral ve kraliçe sözcükleri de benzer şekilde konumlanır. Ayrıca kral vektöründen erkek vektörü çıkarıldığında ve buna kadın vektörü eklendiğinde kraliçe vektörü elde edilmektedir (Vaswani vd., 2023).



Şekil 4: *Katıştırma uzayında sözcük katıştırma vektörleri (Vaswani vd., 2023)*

Şekil 4'te de görüldüğü üzere genelde basit olması ve konunun anlaşılması açısından 2 veya 3 boyutlu olarak gösterilen sözcük katıştırmaları gerçekte yüzlerce boyuta sahip olabilir. Yüzlerce boyutlu katıştırma uzayında yüzlerce boyutlu sözcük katıştırma vektörleri arasındaki uzaklığın öklid uzaklığı ile ifade edilmesi zor olacağı için kosinüs benzerliği (*cosine similarity*) kullanılmaktadır.

2.3 Sözcük Katıştırmaları ile Büyük Dil Modeli Oluşturma

Sözcük katıştırmaları; çok sayıda metinsel ifadenin olduğu külliyatlardan, derlemelerden elde edilen yapılandırılmamış veri (*unstructured data*) kümeleri ile denetimsiz öğrenme yöntemi kullanılarak aşağıda ifade edilen iki tür ana yaklaşımla oluşturulabilir:

- En baştan kendi gereksinimlerine göre sözcük katıştırma modelini oluşturma (custom embeddings)
- Önceden eğitilmiş sözcük katıştırma modellerini (pre-trained word embeddings) kendi gereksinimlerine göre zenginleştirerek oluşturma
    - Statik sözcük katıştırmaları (static word embeddings) kullanarak oluşturma
    - Bağlamlaştırılmış sözcük katıştırmaları (contextualized word embeddings) ile oluşturma.

Sıfırdan başlayarak sözcük katıştırma modelini oluştururken eğitim için kullanılacak metinsel ifadeler geliştiricinin kendi geliştireceği yapay öğrenme modelinde eğitilmektedir. Bu yaklaşımın diğer yaklaşımlara olan üstünlüğü kullanım alanına, kullanım amacına ve veri kümesine özel bir sözcük katıştırma modeline sahip olunacak olmasıdır. Bu yaklaşımın diğer yaklaşımlara olan zorluğu ise hesaplama maliyetinin yüksek olmasıdır. Diğer bir ifadeyle sözcük katıştırmaları oluşturmak için çok fazla sayıda yüksek kapasiteli "Grafik İşlemci Birimi (Graphical Processing Unit, GPU)" kullanarak çok fazla veriyi uzun zaman içerisinde eğitmek gerekmektedir.

Büyük Dil Modelleri güçlüdür, ancak güçleri eğitildikleri veri kümeleri ile sınırlıdır. Ölçeklendirme (*scaling*), genel olarak bir modeli daha fazla işlem gücü ile daha fazla veri üzerinde daha fazla parametre ile eğitip daha büyük hale getirme sürecidir. Dil modelleri, büyük dil modelleri olarak ölçeklendirildiğinde modellerin doğal dildeki örüntüleri öğrenme becerisi de büyük ölçüde artmaktadır. Bu durum büyük bilgi işlem gücü ile önceden eğitilmiş sözcük katıştırma modellerinin üstünlüğü olarak kabul edilebilir.

Önceden eğitilmiş sözcük katıştırma modelleri; statik sözcük katıştırmaları ve bağlamlaştırılmış sözcük katıştırmaları olmak üzere iki farklı yaklaşımla oluşturulabilir. Statik sözcük katıştırmalarında bir dildeki her sözcük için, o sözcüğün o dilde kullanıldığı bütün bağlamlardan yola çıkarak tek bir sözcük katıştırması üretilirken, bağlamlaştırılmış sözcük katıştırmalarında cümlenin bağlamı dikkate alınarak farklı katıştırmalar üretilir. Bu durumda bağlamlaştırılmış sözcük katıştırmalarında aynı sözcüğün farklı iki bağlamdaki cümleden üretilen katıştırmaları farklı olacaktır. Ayrıca statik sözcük katıştırmaları bir sözcük için tek bir katıştırma oluşturduğu için çok anlamlılığı ve eş adlılığı temsil edememektedir. Diğer bir ifadeyle eş adlı veya çok anlamlı sözcüklerin farklı anlamları arasında ayrım yapılamamaktadır. Örneğin "yazın denize gireceğim" ve "eve ulaşınca bana mesaj yaz" tümcelerinde kullanılan hem mevsim hem de eylem olan "yaz" sözcüğü arasında ayrım yapılamamaktadır. Bağlamlaştırılmış sözcük katıştırmalarında böyle bir sorun bulunmamaktadır. Tablo 1 önceden eğitilmiş sözcük katıştırmalarının örneklerini ve bunlara ait bazı bilgileri göstermektedir.

Tablo 1: *Sözcük katıştırmaları ve özellikleri*

Önceden eğitilmiş sözcük katıştırmaları (*Pre-trained word embeddings*)	**Katıştırma türü**	**Eğitimi gerçekleştiren**
Word2Vec	Statik	Google
Global Vectors (GloVe)	Statik	Stanford University
fastText	Statik	Facebook
Enough, Let's Moe On (ELMo)	Statik	Allen Institute for AI (Ai 2)
Universal Sentence Encoders (USE)	Statik	Google
BERT	Bağlamsal	Google
ALBERT	Bağlamsal	Lan vd. (2020)
Bert Sentence Embeddings	Bağlamsal	
XLNET	Bağlamsal	Yang vd. (2020)
XLM	Bağlamsal	Lample ve Conneau (2019)
RoBERTa	Bağlamsal	Liu vd. (2019)
DistilBERT	Bağlamsal	Sanh vd. (2020)
Electra	Bağlamsal	Standford University & Google

Bir dil modeli veya bir sözcük katıştırma modeli üzerinde mimari değişiklikler yapılarak ve/veya uygulama alanına özel veri kümeleri kullanılarak dil modeline denetimli bir şekilde ince ayar yapılabilmektedir. Bu yaklaşımda modelin genel bilgisinin korunması sağlanırken belirli bir alanda uzmanlaşması da sağlanmış olur. Bunun sonucunda gereksinimlere göre özel olarak ayarlanmış büyük dil modeli (*fine-tuned LLM*) elde edilir. Bir büyük dil modelinin ince ayarını yapmak için en çok kullanılan tekniklerden biri "Low Rank Adaptation (LoRA)" tekniğidir. Bu teknikte model parametrelerinin büyük kısmı dondurulur yalnızca küçük bir kısmı yeniden eğitilir. LoRA tekniği Hugging Face kod alanı içerisinde bulunan "Parameter Efficient Fine Tuning (PEFT)" kütüphanesi kullanılarak gerçekleştirilebilir. LoRA tekniğine ek olarak kullanılabilecek diğer bir teknikte "Quantized LoRA (QLoRA)" tekniğidir. Büyük dil modellerini belirli bir alana yönelik özelleştirmenin diğer bir yolu da "Retrieval Augmented Generation (RAG)" yaklaşımıdır. RAG yaklaşımı harici bir veri kaynağındaki veri kümelerinde yararlanarak yanıtların kalitesini arttırmak için kullanılan bir tekniktir. Özellikle büyük dil modellerinin halüsinasyon problemine çözüm olarak geliştirilmiştir.

Bunların dışında diğer bir yaklaşım da istem mühendisliğindeki (*prompt engineering*) bağlam içi öğrenme yaklaşımını uygulamaktır. Bağlam içi öğrenme, sorgu istemine bazı ekstra bilgi parçalarının Chain of Thought (CoT), Tree of Thought (ToT), Skeleton of Thought (SoT), Graph of Thought (GoT), Algortihm of Thought (AoT) vb. istem mühendisliği teknikleri kullanılarak eklenmesidir.

Çok boyutlu vektör olarak üretilen katıştırmaları depolamak için vektör veri tabanları (*vector databases*) kullanılmaktadır. Katıştırmalar yalnız tümceler, sözcükler ve alt sözcükler için oluşturulmamaktadır. Diğer yapılandırılmamış veri kümeleri olan görsel, video ve ses içinde vektör katıştırmaları oluşturulabilir ve vektör veri tabanlarında depolanabilir.

Bir vektör veri tabanı hızlı erişim (*fast retrieval*) ve benzerlik araması (*similarity search*) için vektör katıştırmalarını indeksler ve depolar. Binlerce vektör arasında uzaklık metriğine dayalı bir sorgu gerçekleştirmek son derece yavaş olacağı için yapılandırılmamış veri kümelerini yalnızca katıştırma olarak depolamak yeterli değildir. Bu nedenle vektörlerin indekslenmesi de gerekir. Dolayısıyla indeksleme işlemi bir vektör veri tabanının ikinci kilit unsurudur. Arama işlemini kolaylaştıran bir veri yapısı olan indeksleme için vektör veri tabanlarında Hierarchical Navigable Small-World (HNSW), DiskANN (Subramanya vd., 2019) gibi faklı indeksleme yöntemleri kullanılmaktadır. Tablo 2'de görüldüğü üzere farklı özellikli birçok vektör veri tabanı bulunmaktadır.

Tablo 2: *Vektör veri tabanları ve özellikleri*

Vektör veri tabanı	Kaynak kod	Programlama dili	İndeks yöntemi
Vespa	Açık	C/C++	NHNSW + BM25 hibrit
Weaviate	Açık	Go	HNSW, özelleştirilmiş HNSW, DiskANN
Milvus	Açık	Go	IVF, HNSW, RHNSW, DiskANN
Zilliz	Kapalı	Go	IVF, HNSW, RHNSW, DiskANN
Vald	Açık	Go	NGT
drant	Açık	Rust	özelleştirilmiş HNSW
Pinecone	Kapalı	Rust	özel karmaşık index
Elastic Search	Açık	Java	HNSW
Redis	Açık	C/C++	HNSW
Chroma	Açık	C/C++	HNSW
LanceDB	Açık	Rust	IVF, DiskANN
pgvector	Açık	C/C++	IVF

3. Büyük Görsel Modeller

Difüzyon modelleri (*diffusion models*), görüntü veya ses üretimi gibi birçok farklı alanda kullanılan üretici modellerdir (*generative models*). Difüzyon modelleri, koşullandırma olmadan rastlantısal bir görüntü oluştururken, metinsel ifadeden görüntü (*text-to-image*) oluşturan modeller koşullandırma yaparak girilen metin istemine uygun bir görüntü üretmektedir. Metin istemlerinden görüntüler oluşturabilen veya bir metin istemiyle mevcut görüntüleri değiştirebilen açık kaynak kodlu olarak Stable Diffusion (StabilityAI – 2022) ve kapalı kaynak kodlu olan DALL-E (OpenAI – 2021), Midjourney (bağımsız geliştiriciler – 2022) vb. üretken yapay zekâ uygulamaları bulunmaktadır.

Gerçekçi görünen sentetik görüntüler oluşturan derin üretici ağ modelleri arasından en öne çıkanı 2014 yılında Goodfellow vd. tarafından geliştirilmiş olan Çekişmeli Üretici Ağlar (Generative Adversarial Networks, GANs)'dır (Goodfellow vd., 2014). Sonrasında farklı araştırmacılar tarafından CycleGAN, BiCycleGAN, ReCycleGAN, StyleGAN, Fast StyleGAN, pixelRNN, text-2-image, DiscoGAN, lsGAN gibi birçok farklı GAN mimarisi ortaya çıkarılmıştır.

Nvidia araştırmacıları 2018 yılında yüksek çözünürlüklü sentetik görüntü oluşturan Style-Based Generator Architecture for Generative Adversarial Networks (StyleGAN) mimarisini duyurmuşlar ve kaynak kodunu da yayınlamışlardır. StyleGAN yüz sentezi yapabilen bir yapay öğrenme algoritmasıdır. Genel olarak bu yapay öğrenme algoritması bir insanın yüz

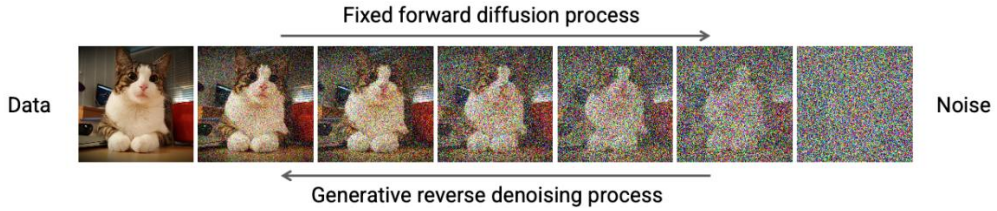
hareketlerini öğrenmekte ve bunu tarihi bir figüre uyarlamaktadır. Özellikle Mona Lisa gibi tarihi figürlere gerçek insanların yüzlerinin sentezlenmesi ile popülerlik kazanmıştır.

DeepFake uygulaması da benzer yaklaşımı görüntü yerine video üzerine uygulamaktadır. Bu videolarda genellikle bir kişinin videosu üzerinde oynamaya gidilerek video içeriği bir başka kişinin görüntüsü ile değiştirilmektedir. DeepVoice uygulaması da benzer tekniği ses üretim için gerçekleştirilmektedir.

3.1 Difüzyon Modellerinin Çalışma Şekli

Metin istemlerinden görüntü oluşturan difüzyon modelleri genel olarak aşağıda ifade edildiği ve Şekil 5'te gösterildiği gibi çalışmaktadır:

- **Veriden gürültüye ileri besleme:** Bir dizi eğitim görüntüsüne görüntü yalnız gürültüden oluşana kadar rastlantısal gürültü (*random noise*) eklenir.
    - Birçok farklı gürültü türü olmasına rağmen difüzyon modellerinde normal dağılım özelliği olan Gauss gürültüsü (*Gaussian noise*) kullanılır. Bir görüntüye Gauss gürültüsü eklemek, o görüntünün piksel değerlerini ve olasılık dağılımını biraz değiştirmek anlamına gelmektedir.
    - Gürültü, bir Markov zincirini takip ederek görüntülere uygulanır. Diğer bir ifadeyle, her zaman adımında görüntüye bir parça gürültü eklenir.
- **Gürültüden veriye geri besleme:** Metin istemine uygun görüntüyü oluşturmak için eklenen gürültünün nasıl tersine çevrileceği (*denoising*) öğrenilir. Böylece model yüksek çözünürlüklü görüntüler üretebilir.
    - Gürültünün tersine çevrilmesi için evrişimsel sinir ağları (*convolutional neural network*) kullanılarak görüntü piksellerinin değerleri geri kazandırılmaya çalışılır. Diğer bir ifadeyle yapay sinir ağı gürültüyü tahmin etmeyi öğrenir.
    - Nichol vd. (2021), evrişimsel sinir ağı olarak UNet kullanmıştır. Konvolüsyonlar aracılığıyla görüntünün küçük bir temsilini oluşturması ve ardından orijinal boyutlara geri örneklemesi nedeniyle bu şekilde adlandırılmaktadır. Bu şekilde ağların giriş ve çıkış boyutları aynı boyuta sahip olmaktadır (Nichol vd., 2021).



Şekil 5: *Difüzyon modeli çalışma şekli (URL-1)*

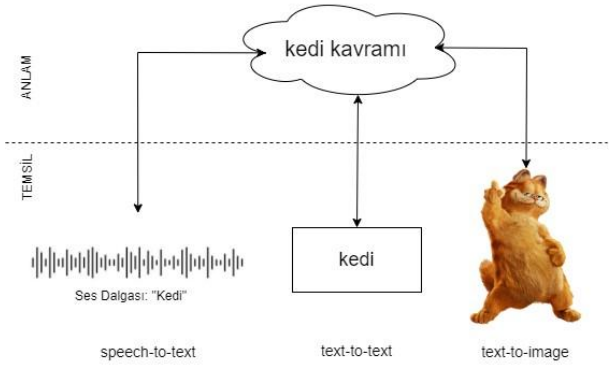
4. Çok Modlu Üretken Yapay Öğrenme

Kısa zaman içerisinde tek modda (*single modal*) dönüşüm (*transform*) yapan model ve uygulamalar yaygınlaşmıştır. Büyük dil modeline soru sorup yanıt alma (*text-to-text*) tek modlu dönüşüme örnektir. Bu yaklaşımda akıl yürütme ve çıkarım gerçekleştirilmiştir. Bir sonraki adımda çok modlu görevleri (*multi-modal tasks*) yerine getirebilen ChatGPT, Gemini gibi yaklaşımlar ortaya çıkmıştır. Son olarak üretken yapay zekâ, Figure AI gibi insansı robotlarla (*humanoid robots*) bütünleştirilmeye başlanmıştır.

Metin ve görüntü biçiminde her iki mod aynı anlamsal kavramı (*semantic concept*) temsil edebilir. Şekil 6'da görüldüğü

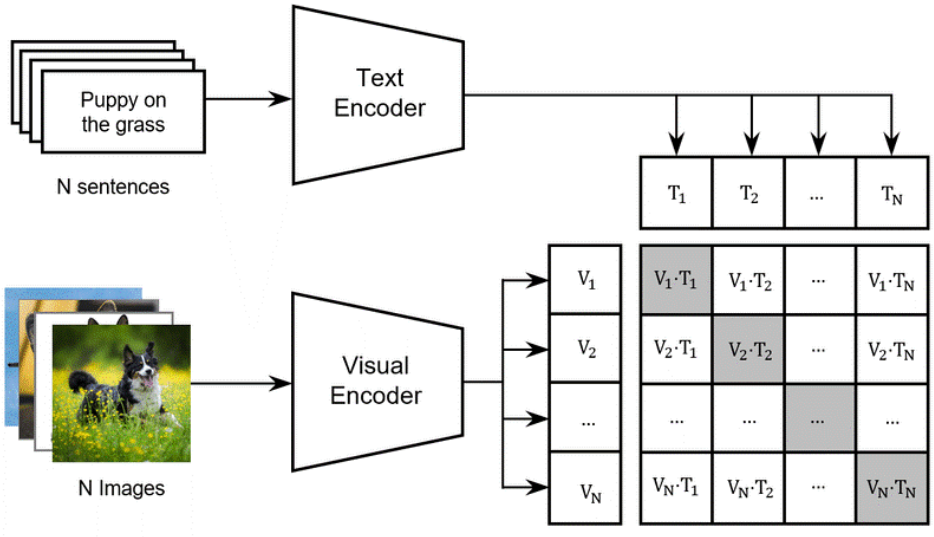
üzere bir insan, kedi sözcüğünü gördüğünde de kedi görüntüsünü gördüğünde de kedi ifadesini duyduğunda da aynı temel anlamı yani kedi kavramını anlar. Tek fark birinin metin (*textual representation*) birinin görsel (*visual representation*) diğerinin ses (*audio representation*) olarak temsil edilmesidir. Metin, görüntü, video, ses, kod, vb. bunlar birer temsildir. Aynı zamanda birer modalitedir (*modalities*). Üretken yapay öğrenme uygulamaları metin, görüntü, video ve ses gibi çeşitli veri türlerini içeren çok modlu bir yaklaşıma (*multi-modal approach*) doğru evrilmiştir.

Metinsel ifadeden görüntü üreten modeller (*text-to-image models*), girişte metin çıkışta görüntüler olmak üzere modelin her iki ucunda yalnızca bir modu (*one modality*) işlemektedir. Bu yaklaşım diğer modların eklenmesi ile zenginleştirilebilir ve böylece çok modlu (*multiple modalities*) yaklaşıma geçilmiş olur. Çok modlu bir yapay öğrenme modeli (multimodal AI) girdi olarak metin, görüntü, video ve ses gibi faklı modları alabilir ve çıktı olarak metin, görüntü, video ve ses gibi farklı modları üretebilir. Örneğin bir üretici yapay zekâ uygulamasından konuşarak bir kedi görüntüsü oluşturması istenilirse ses önce metne (*speech-to-text*) dönüştürülecek, ardından büyük dil modeli bu istemi yorumlayacak (*text-to-text*) ve son olarak bu yorum görüntü üretimine (text-to-image) dönüşecektir. Bu örnekte ilk bakışta konuşma/ses girdisinden doğrudan görüntü elde edildiği zannedilse de aslında sesten görüntü (*speech/audio-to-image*) üreten bir model bulunmamaktadır. Onun yerine yapay öğrenme tabanlı doğal dil işleme yetenekleri olan büyük dil modelleri metin modu üzerinden farklı modları birbirine bağlamak için kullanılmaktadır. Başka bir ifadeyle çok modlu yapay öğrenmede modelin merkezini diğer bir ifadeyle düşünme mekanizmasını büyük dil modeli oluşturmaktadır.



Şekil 6: *Aynı kavramın farklı modlarla temsili*

Çok modlu bir yapay öğrenme modeli, doğrudan bu farklı temsillerin kendi temsil yapılarıyla çalışmak yerine anlam (*meaning*) üzerinden çıkarımlar yapmaktadır. Çok modlu modeller, metin, görüntü gibi modların anlamlarını yakalayan katıştırmaları kullanmaktadır. Başka bir ifadeyle metin, görüntü, ses gibi farklı temsiller vektör temsiline dönüştürülerek kullanılmaktadır. Örneğin DALL-E gibi bir difüzyon modeli katıştırmaları eğitmek ve dolayısıyla anlam uzayını (*meaning space*) öğrenmek için CLIP modelinden yararlanır. CLIP gibi modeller altyazılı görüntülerden eğitilir. Alt yazılardan metin kodlayıcılar (*text encoders*), görüntülerden görüntü kodlayıcılar (*image encoders*) oluşturulur. Daha sonra farklı modda bulunan bu çiftlerin kosinüs benzerliği üzerinden eğitim aşamasına geçilir. Aynı kavram için benzerlik arttığında metin ve görüntü vektörleri yakınlaşacak yani kosinüs benzerliği artacak, iki farklı kavram için metin ve görüntü vektörlerinin kosinüs benzerliğini en aza indirecek şekilde eğitim gerçekleştirilir. Bu işlem veri kümesindeki her metin ve görüntü vektörü kombinasyonu için gerçekleştirilir. Ardından görsel anlam vektörünün bir görüntü için kodu çözülür (*decoding*) ve difüzyon modeli görüntüyü oluşturur.



Şekil 7: *Çok modlu yapay öğrenme modelinin çalışma şekli (Song vd., 2022)*

Google firması 6 Aralık 2023 tarihinde AlphaGo gibi planlama algoritmalarıyla büyük dil modellerinin bir kombinasyonu olan Gemini çözümünü çok modlu yapay öğreme modeli olarak duyurmuştur. Tanıtım videolarında gösterildiği gibi Gemini gerçek zamanlı çalışmamaktadır. Sözü edilen tanıtım videolarında şu ifade yer almaktadır: "*ara işlemler atlanarak tanıtım videoları oluşturulmuştur*". Ancak yine de Gemini çok modlu yapay öğrenme modelinin olabileceğini göstermiştir.

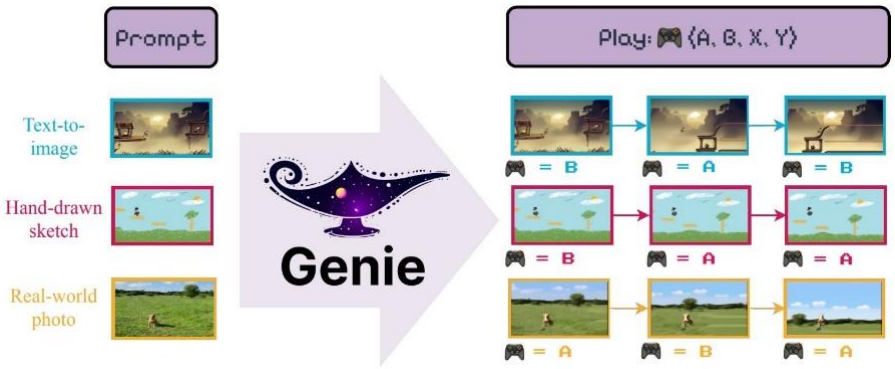
OpenAI firması tarafından 15 Şubat 2024 tarihinde metinsel ifadeden video üreten (*text-to-video*) SORA uygulamasını yayınlanmıştır. Pika.art, runway, Leonardo AI, StableDiffusion, Lumiere gibi daha önce de metinsel ifadeden video üreten uygulamalar olsa da SORA uygulamasının akıcılığı, gerçekçiliği, yüksek çözünürlüğü ve en önemlisi fizik kurallarına uygun çalışması farklılık oluşturmaktadır. OpenAI firması SORA çözümü için "*Sora, gerçek dünyayı anlayabilen ve benzetim yapabilen modeller için bir temel olacaktır*" ifadesini kullanmıştır. Bu açıdan bakıldığında SORA çözümünün gerçek dünya problemlerinin çözümünde kullanılabilecek bir araç olarak kabul edilebileceği düşünülebilir.

Google firması 15 Şubat 2024 tarihinde bir milyon token ile çalışabilen Gemini Pro 1.5 çözümünü duyurmuştur. Aynı zamanda Gemini 1.5 sürümüyle birlikte üretken yapay öğrenme modelleri video modunu girdi olarak alıp çıkarım yapmaya başlamıştır. Daha önceki videodan metin çıkarma uygulamaları (*video-to-text*) videonun kendisini izleyerek değil video içerisindeki transkriptleri okuyarak yorumlama yapmaktaydı. Artık gerçek dünya problemlerine yönelik SORA tarafından üretilecek simülasyon videoları Gemini 1.5 tarafından yorumlanarak farklı modlara dönüştürülebilir.

Gerek metinsel ifadeden görüntü (*text-to-image*) üreten DALL-E, Midjourney, StableDiffusion gibi uygulamaların oluşturduğu görüntülerin, gerekse metinsel ifadeden video (*text-to-video*) üreten SORA gibi uygulamaların oluşturduğu videoların gerçek dünya problemlerinin çözümü için etkileşimli bir şekilde kullanılabilmeleri gerekmektedir. Örneğin İstanbul gibi bir kentin dijital ikizi temel alınarak 3B kent modelinin SORA uygulaması üzerinde ulaşım, otonom sürüş, enerji, afet gibi tematik alanlarda simülasyonu yapıldığı düşünüldüğünde bu benzetimlerle etkileşimin nasıl kurulabileceği, benzetim üzerindeki senaryoların ve kullanım durumlarının (*use-cases*) nasıl değiştirilebileceği bir diğer sorun olarak ortaya çıkmaktadır.

Google firması 23 Şubat 2024 tarihinde Generative Interaction Environment (GENIE) uygulamasını duyurmuştur. GENIE web üzerinden bulunan etiketi olmayan videolarından denetimsiz olarak eğitilmiş bir temel dünya modelidir (*foundation world model*). Şekil 8'de görüldüğü üzere GENIE; gerçek görüntülerden, sentetik görüntülerden (*text-to-image*), el

çizimlerinden sonsuz çeşitlilikte oyun oynanabilir yani etkileşimle kontrol edilebilir dünyalar oluşturabilmektedir (Bruce vd., 2024). Yine benzer biçimde SORA tarafından üretilecek simülasyonların GENIE içerisinde etkileşimli biçimde kullanılabilir olması önemli bir gelişme olacaktır.



Şekil 8: *GENIE çalışma prensibi (Bruce vd., 2024)*

Google firması 13 Mart 2024 tarihinde Scalable Instructable Multiworld Agent (SIMA) çözümünü duyurmuştur. SIMA 3B sanal ortamlar için eğitilmiş üretken bir etmendir. SIMA içerisinde etmenler farklı ortamlarda (farklı oyunlarda) eğitilebilir ve tüm bu eğitimlerde kazanılan yetenekler farlı bir simülasyon ortamında kullanılabilir.

Bununla birlikte Github Copilot, Goose, Blackbox AI, Devin AI gibi metinsel ifadeden kod (*text-to-code*) üretimi için özelleştirilmiş birçok uygulama ortaya çıkmaktadır. Cognition Labs firması 12 Mart 2023 tarihinde yapay öğrenme modeli eğitebilen Daven AI çözümünü duyurulmuştur. Böylece örneğin Hugging Face kod alanı içinde bulunan daha önceden SORA, GENIE vb. kullanılarak üretilen İstanbul kentinin yapay öğrenme modelinin beni oku (*readme*) sayfası Daven AI uygulamasına verilerek Ankara gibi başka bir kent için 40 milyar gibi milyarlarca parametreli bir kent modeli eğitilebilir.

Yapay genel zekâ türünde her şeyi yapabilen Jarvis, Skynet gibi tek bir yapay zekâ modeli yerine farklı şeyleri yapabilen birçok yapay öğrenme modeli bulunmaktadır. Bu durumda örneğin SORA uygulamasında üretilen bir videonun GENIE uygulamasında kontrol edilebilmesi gibi farklı kişiler veya kuruluşlar tarafından geliştirilen üretken yapay öğrenme modellerinin birlikte kullanımı sorunu ortaya çıkmaktadır. Sözü edilen modeller arasında bu tür bir birlikte çalışabilirliğin kurulması; İstanbul'un jeodezik ölçülere dayalı oluşturulmuş dijital ikizinin fizik kurallarına dayalı olarak oluşturulacak uzamsal modeli ile etkileşiminin, uzamsal analizlerinin ve karar verme süreçlerinin etmenler tarafından yapılabilmesini olanaklı hale getirecektir.

Büyük dil modellerini yalnız sözcüklerle değil de görüntü, video, ses gibi diğer temsillerle birlikte eğitildiği düşünüldüğünde "Genel Dünya/Gerçeklik Modeline (General World Model, GWM)" biraz daha yaklaşılmış olacaktır. Diğer ifadeyle çok modlu yapay öğrenme ile dünyanın işleyişinin anlaşılması, etrafında olup bitenler arasında bağlantıların kurulabilmesi, buradan kendini daha da geliştirebilmesi mümkün olacaktır. Aslında otonom sürüş alanında sözü edilen genel dünya modeli yaklaşımına benzer bir yaklaşım kullanılmaktadır. Otonom araç üzerindeki sensörlerden farklı modlarda veri üretilmekte ve bu farklı veri kümeleri otonom sürüş kararları ve eylemeleri için birlikte kullanılmaktadır.

4.1 Çok Modlu Öğrenmenin Çalışma Şekli

Çok modlu üretken yapay öğrenme modeli yaklaşımı aşağıda ifade edilen üç ana bileşenden oluşmaktadır (Wu S. vd., 2023):

- **Çok Modlu Kodlayıcı (*Multimodal Encoding*):** Metin olmayan girdileri metin istemlerine dönüştürme görevini yerine getirmektedir. Her farklı moddaki her bir girdinin anlamsal katıştırması (*semantic embedding*) oluşturulur.

Ardından bu katıştırmalar büyük dil modelinin kullanabilmesi için metinsel temsile dönüştürülür.

- Çok modlu kodlayıcı kısmında Meta firmasının açık kaynaklı ImageBind çoklu ortak katıştırma modeli (*multi joint embedding model*) kullanılabilir.
- ImageBind birden fazla modu işleyebilen ve aynı katıştırma uzayında anlamsal katıştırmalar üretebilen bir modeldir.

- **Büyük Dil Modeli:** Metin temsillerini anlayarak ve bunlar üzerinde akıl yürüterek kodlayıcı bileşeninden gelen metinsel ifadelere yanıt verme ve çıktı için diğer modların oluşturulmasına ilişkin yönlendirme yapma görevlerini yerine getirmektedir. Farklı modların birbirine karışmadan üretim yapmasını sağlamak için yönlendirmelerinde token kullanmaktadır.
- **Çok Modlu Kod Çözücü (*Multimodal Decoding*):** Büyük dil modeli bileşeninden gelen ayrı ayrı modlarla ilgili olan çıktıların dönüştürücü (*transformer*) tabanlı modellerle hazırlanması görevini yerine getirmektedir. Böylece difüzyon kod çözücülerle çok modlu çıktı üretilmektedir.

Birinci bölümde büyük dil modellerinin amaca yönelik olarak özelleştirilebileceğinden (*custom LMM*) bahsedilmiştir. Bu bölümde de çok modlu üretken yapay öğrenme modellerinden ve bu modellerin merkezinde büyük dil modellerinin yer aldığından söz edilmektedir. Bir diğer ifadeyle dil modelleri kavramı "Çok Modlu Büyük Dil Modelleri (Multimodel Large Language Model, MMLLM)" olarak genişlemiştir. Kurum ve kuruluşların büyük dil modellerinde olduğu gibi çok modlu büyük dil modellerini etkin kullanabilmeleri için belirli bir alana veya işlevselliğe özel olarak eğitmeleri gerekmektedir. Bu durum da "Özelleştirilmiş Çok Modlu Büyük Dil Modelleri (customized MMLLM veya custom multimodal AI)" kavramını ortaya çıkarmaktadır. Bu özelleştirme, modellerin daha yüksek doğrulukta ve bağlam içinde sonuçlar üretmesini sağlayacaktır. Kurum ve kuruluşların bu modelleri iş akışlarına doğrudan entegre edilebilmeleri için önceden geliştirmiş oldukları mikro servislerle birlikte kullanabilmeleri gerekmektedir. Dijital dönüşümünü tamamlamış kurum ve kuruluşların sunucularında bulunan veri tabanları üzerindeki yapılandırılmış veri kümeleri ile bu modelleri kullanabilmeleri gerekmektedir.

4.2 Büyük Eylem Modelleri

"Büyük Eylem Modelleri (Large Action Model, LAM)" veya "Büyük Etmen Modelleri (Large Agentic Models, LAMs)" büyük dil modellerinin bir uzantısıdır. Büyük eylem modeli, kullanıcının ona verdiği istemelere göre gerçek zamanlı olarak eyleme geçebilen bir modeldir. Yemek siparişi vermek, taksi çağırmak, hotel rezervasyonu yapmak vb. eyleme dayalı işler büyük eylem modeli ile gerçekleştirilebilir. Eyleme dönüştürme görevi gören büyük eylem modelleri ile üretken yapay zekânın tam olarak kişisel asistan olması hedefi gerçekleşmiş olacaktır.

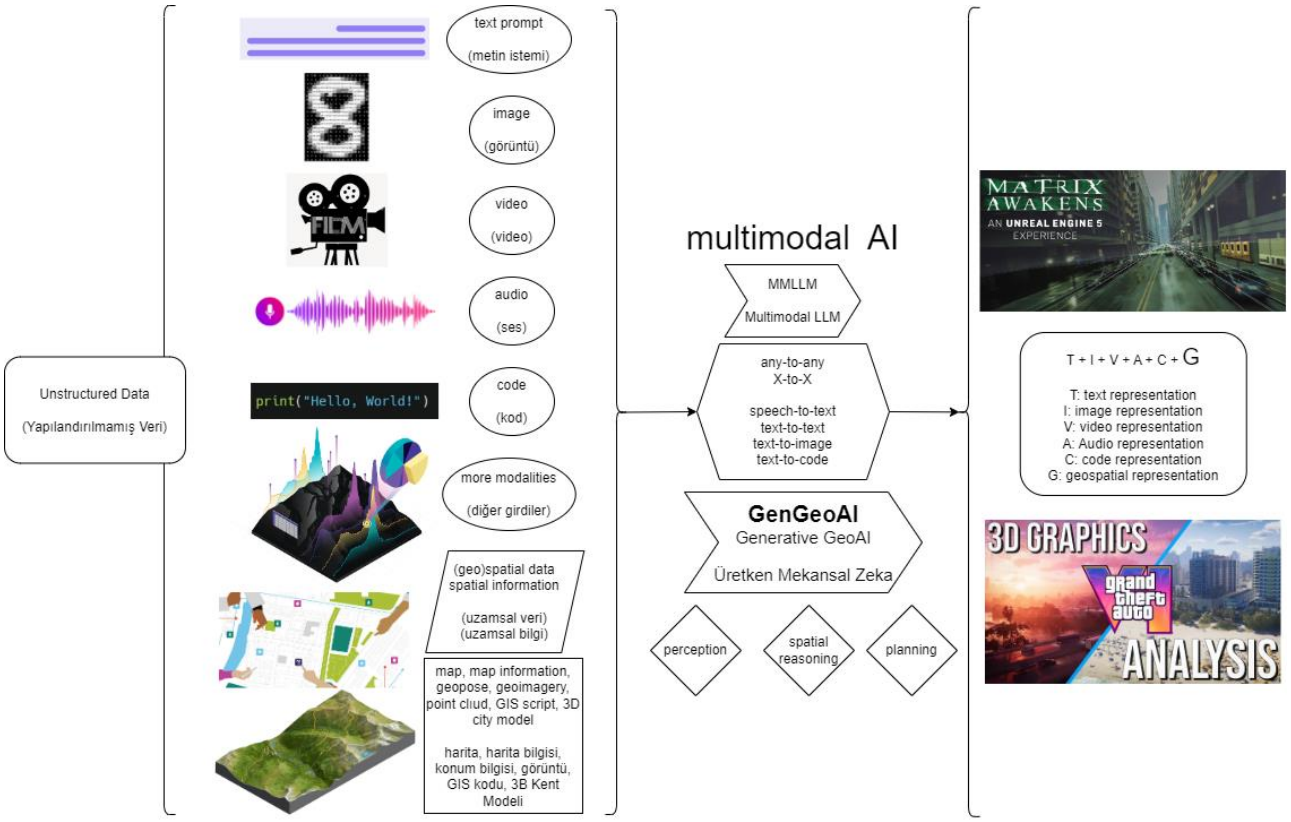
Etmenler kendi başlarına görevler yürütebilen yazılım birimleridir. Bu nedenle büyük dil modellerinde olduğu gibi insanın sorularını/istemlerini yanıtlamak yerine, bir görevi yerine getirirler. Büyük eylem modelleri yaklaşımındaki etmenler; nesnelerin interneti (IoT) cihazları, otomatik cihazlar ve uygulama programlama arayüzleri (API) gibi harici sistemlerle işlevsel bir iş birliği kurarak gerçek dünya ile etkileşime girebilir. Büyük eylem modelleri bu tür cihaz ve uygulamalara bağlanarak fiziksel eylemler gerçekleştirebilir, cihazları kontrol edebilir, JSON/GeoJSON dosyası olarak veri alabilir veya bir hesap tablosu üzerinde bilgi işlem görevin yerine getirebilir.

4.3 Üretken Uzamsal Öğrenme

Çok modlu yapay öğrenme modellerinde uzamsal bilgi (spatial information) de bir mod olarak kullanılabilir. Üretken yapay zekâ kavramı temel modeller (foundation models) üzerinde geliştirilmektedir. Metin, görüntü gibi farklı temsiller için

önceden geliştirilmiş temel modeller bulunmaktadır. Nasıl metinsel ifadeler için büyük dil modelleri varsa, uzamsal (mekânsal, coğrafi) ifadelerin de yer alacağı uzamsal etkin büyük dil modelleri (Geo enable LLM veya GeoLLM) ince ayar yapılarak çok modlu üretken yapay zekâ uygulamaları Şekil 9'da gösterildiği gibi zenginleştirilebilir. Daha sonra da GeoGPT gibi uygulamalar GeoLLM modelleri üzerine geliştirilebilir. Bu tür uygulamalara Geo+ ve GeoForge örnek olarak verilebilir. Böylece aşağıdaki gibi mekânsal analizler yapılabilir:

- Sarıyer İlçesi Maslak mahallesinde yapılacak olan içme suyu hattındaki arıza çalışması için hangi vanalar kapatılmalıdır? Buna bağlı olarak hangi binalar susuz kalacaktır? Hangi abonelere SMS gönderilecektir? Bu tür sorular üretken uzamsal uygulamasına istem olarak girilebilir.
- Kapatılacak vanaların yerlerinin haritada veya 3B modelde gösterilmesi istenebilir. Vanaların konum bilgilerini GeoJSON veya CSV dosya türünde bir tablo olarak dışarıya aktarılabilir.
- İstanbul ili Sarıyer İlçesi Maslak Mahallesi için Sentinel 2 uydu görüntülerinden NDVI gibi indeksler oluşturması istenilebilir.



Şekil 9: *Üretici uzamsal zekâ modeli*

İstanbul'da deprem durumunda afet anında barınma, lojistik gibi sorunların çözümü için mekânsal/coğrafi bilgi sistemi oluşturulmak istense iş adımları genel olarak aşağıdaki gibi olacaktır:

- Verinin elde edilmesi
    - Afet anı öncesi kentin topografik haritası
    - Afet anı sonrası uydu görüntüleri, fotogrametrik görüntüler
        - Görüntülerden yapay öğrenme algoritmaları ile hasarlı yerlerin, yıkılan binaların, kapalı yolların vb. bilgilerin çıkarılması
    - Sosyal medya uygulamaları üzerinden veri kazıma yapacak yapay öğrenme algoritmaları ile bilgi çıkarılması

    - Ulaşım için güncel yol ağı ve yol ağı topolojisi
    - Lojistik yerlerin ilgi noktaları
    - Nerede ne kadar insan yaşadığı
    - Diğer veri kümeleri
- Verinin veri tabanları üzerinden sistematik hale getirilmesi ve veri tabanı modeli üzerinden ilişkilendirilmesi
- Verinin analiz edilmesi
    - Mekânsal/Coğrafi sorgulamaların yapılması
    - Mekânsal/Coğrafi analizlerin gerçekleştirilmesi
- Verinin karar destek için kullanılması
    - Karar destek için algoritmaların uygulanması
- İlgili tematik haritaların üretimi

Çok modlu üretken uzamsal yapay öğrenme ile yukarıdaki senaryo yalnız istemler girilerek gerçekleştirilebilir. Örnek istemler aşağıda belirtilmiştir:

- Kamu kurum ve kuruluşları için örnek istem: "Kentlilerin/afetzedelerin yaşanan son deprem sonrasında ulaşabilmeleri için İstanbul'un geneline yönelik mevcut enkaz durumlarında kullanımı en uygun olabilecek barınma ve lojistik alanlarının yerlerini belirle."
- Afetzede olarak örnek istem: "Bulunduğum konuma en yakın geçici/kalıcı barınma yerlerini ve lojistik merkezlerini otomobille ve yürüyerek ulaşım bilgilerini de içeren bir harita üzerinde göster."

Bu istemlerin gerçekleştirilmesinde herhangi bir insan müdahalesi olmadan bağlantılı veri ve API'ler üzerinden yapay öğrenme modeli ilgili veri kümelerine ulaşabilir, mekânsal/coğrafi analizleri gerçekleştirebilir, karar verme süreçlerini ilerletebilir ve sonuç olarak isteme uygun tematik haritaları üretebilir. Böylelikle ilk senaryoda uzun zaman alacak çözüm ikinci senaryoda çok kısa zamanda üretilebilecektir.

Çok modlu üretken yapay öğrenme uygulamalarının temelinde bulunan büyük dil modellerinin bağlam pencerelerine (*context window*) ilişkin token sayıları arttıkça üretken yapay öğrenme ile uzamsal yapay öğrenme arasındaki boşluk üretken modeller tarafına doğru daha da kapanacaktır. Örneğin Google Gemini 1.5 bir milyon tokena sahiptir ve on milyon tokena yükseltilmesi planlanmaktadır. Bu durumda bugün görece küçük boyutlu görüntülerle gerçekleştirilen üretken uygulama yer gözlem platformlarının farklı sensörlerinden elde edilecek büyük boyutlu görüntüler içinde gerçekleştirilebilecektir. Görüntü üreten uygulamaların API'leri mekânsal bilişim alanında kullanılan CAD/GIS/BIM yazılımlarında etkin olarak kullanılmaya başlanacaktır. Geomatik/Harita mühendislerinin arazide kullandığı uzamsal veri üreten ölçme cihazları hem büyük dil modellerinin hem de büyük eylem modelleri ile bütünleşmeye başlayacaktır. Bu ve benzeri gelişmeler arazide ve/veya ofis ortamında çalışan personel sayısını da etkileyecektir. Büyük eylem modelleri ile destekli ölçme cihazları arazide gerekli personel sayısının azalmasına, büyük dil modelleri destekli mekânsal bilişim yazılımları da ofis ortamındaki personel sayısının azalmasına neden olacaktır.

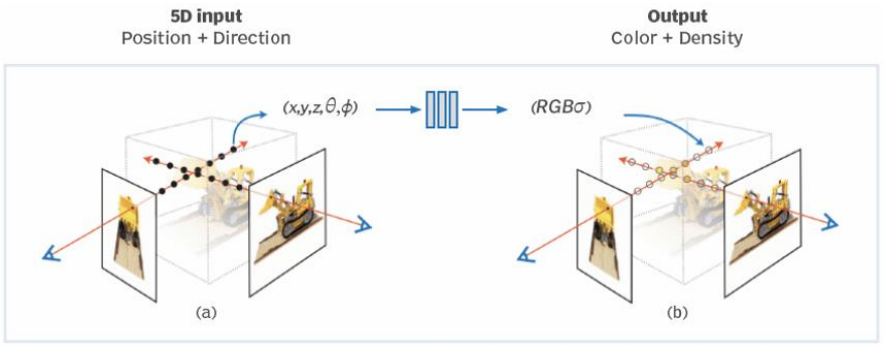
Uzamsal yapay öğrenme modelleri genel olarak gözetimli ve gözetimsiz öğrenme üzerinde yapılanmaktadır. Çok modlu üretken yapay öğrenme uygulamalarında pekiştirmeli öğrenme yaklaşımının kullanımı giderek artmaktadır. Pekiştirmeli öğrenme yaklaşımının uzamsal yapay öğrenmeye yansıması büyük bir paradigma değişimine neden olacaktır. Bu paradigma değişiminin en iyi örneği olarak sürücüsüz araçlarla otonom sürüşün gerçekleştirilmesi gösterilebilir.

4.4 3B Veri Üretimi ve Üç Boyutlu Temsil

Birden fazla bakış açısından elde edilen görüntüler veya videolar kullanılarak gerçek zamanlı, gerçekçi (*photorealistic*) ve yüksek kaliteli sahnelerin oluşturulmasında (*3D reconstruction*), render edilmesinde, temsilinde (*representation*) ve sentezinde (*view synthesis*) önemli ilerlemeler olmuştur. Bu konuda "Sinirsel Işınım Alanı (Neural Radiance Field, NeRF)" ve "3B Normal Dağılımlı Düzensiz Şekillendirme (3D Gaussian Splatting, 3DGS)" olmak üzere iki yöntem öne çıkmaktadır.

4.4.1 Sinirsel Işınım Alanı (NeRF)

NeRF yönteminin amacı yapay sinir ağlarını kullanarak 2B görüntülerden bir sahneyi (*scene*) 3B modellemek ve 3B görsel temsilini (*3D visual representation*) oluşturmaktır. Bu yöntemde üretilen görsel temsil lokal bir koordinat sistemindedir. NeRF yöntemi ile sahne sürekli bir fonksiyon olarak modellenir. Şekil 10'da görüldüğü üzere bu fonksiyon 3B Kartezyen koordinatlar şeklinde görüntü elde etme anındaki kamera konum bilgisini (X, Y, Z), kamera bakış açılarını (Θ, ϕ) girdi olarak kullanıp sahnenin bulunduğu uzaydaki piksellerin görünüme bağlı olarak renklerini (RGB) ve konum bilgisine bağlı olarak ışığın yayılma özelliğini (*radiance*, σ) tahmin etmektedir (Mildenhall vd., 2020).



Şekil 10: *NeRF eğitiminin çalışma şekli (Mildenhall vd., 2020)*

Başka bir ifadeyle 3B sahnenin içinde bulunduğu uzaydaki her nokta için ışık ışınlarının yayılma özelliğini yapay sinir ağı kullanarak hesaplayan bir fonksiyon oluşturulmaktadır. Böylece 2B görüntülerden 3B sahnenin temsili öğrenilmiş olur. Fonksiyon, öğrenilen uzayı yeterince temsil ettiğinde, görüntülerin elde edildiği uzayın tüm 3B bilgisini temsil etme yeteneğine sahip hale gelir.

Bilgisayar grafiği alanında kullanılan bir sahnenin 3B render sürecinde (*volume rendering*) önce şekiller (*shapes*) yani nesnelerin geometrileri mesh veya voksel gibi 3B poligonlarla oluşturulur sonra doku (*texture*) ve malzeme (*material*) efektleri uygulanır. NeRF yöntemi ise geometriyi, dokuyu, malzeme efektini doğrudan karakterize eden ışık alanları (*light fields*) üzerinde çalışmaktadır. Diğer bir ifadeyle nesnelerin gerçekte nasıl farklı görülebildiklerini temsil edebilmektedir.

NeRF yönteminin eğitim aşamasında bir sahneye ilişkin farklı bakış açılarından elde edilen görüntüler ve/veya videolar kullanılır. Görüntülerin veya videolardaki çerçevelerin (*frame*) görüntü elde etme anındaki kamera konum ve bakış açıları bilgileri elde edilir. Daha sonra yukarıda sözü edilen fonksiyon, yapay sinir ağı modeli kullanılarak oluşturulur. Yapay sinir ağı tarafından tahmin edilen renkler ve ışığın yayılma özelliği bilgisi kullanılarak oluşturulan görüntülerle gerçek görüntüler arasındaki benzetim düzeyi kullanılarak yapay sinir ağı eğitilir. Belirli bir düzeyde eşleşme sağlandığında eğitim süreci sonlandırılır. Böylece üretken yapay öğrenme modeli tüm sahneyi temsil edebilir (*scene representation*) hale gelir. Diğer bir ifadeyle artık bu model;

- sahnenin tüm bilgisinin 3B olarak oluşturulması için kullanılabilir (*text-to-3DModeling*)
- sahnenin gerçek görüntüsü olmayan yerlerine ilişkin görüntü oluşturmak için kullanılabilir (*view synthesis*)

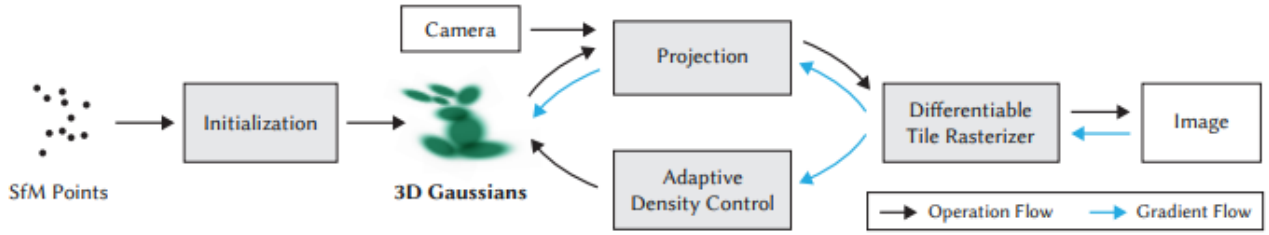
- yeni bakış açılarından gerçekçi görünümler oluşturmada kullanılabilir (*text-to-reality*)
- aynı sahnenin günün farklı saatlerindeki farklı aydınlık koşullarına göre görüntüleri oluşturulabilir.

2020 yılında Mildenhall vd. tarafından sunulan çalışmada görüntüler aynı ışık koşullarında aynı kamera ile elde edilmiş ve "Çok Katmanlı Algılayıcılar (Multilayer Perceptron, MLP)" kullanılarak hesaplama yapılmıştır (Mildenhall vd., 2020). 2022 yılında Nvidia firması tarafından geliştirilen Instant NeRF ve Google firması tarafından geliştirilen NeRF in the Wild gibi ilerlemelerle birlikte farklı kameralarla farklı ışık koşullarında elde edilen görüntülerle farklı derin öğrenme mimarileri uygulanarak kısa süre içerisinde farklı görünümler oluşturulabilmiştir. Sonrasında NeRF yaklaşımı kullanan PixelNeRF (CVPR 2021), Mega-NeRD (CVPR 2022), Neural Sparse Voxel Fields (NSVF), Plenoptic Voxel (2021), Pyramidal Neural Radiance Field (PyNeRF), Mega-NeRF gibi birçok farklı çatkı geliştirilmiştir (Rabby ve Zang, 2024).

4.4.2 3B Normal Dağılımlı Düzensiz Şekillendirme

Gaussian Splatting (GS) geleneksel bir bilgisayar grafiği tekniğidir. Genel olarak hacim oluşturmada (*volume rendering*) ve nokta tabanlı render işlemlerinde kullanılmaktadır (Zwicker vd., 2002).

3D Gaussian Splatting (3DGS) yönteminde Şekil 11'de görüldüğü üzere 3B nokta bulutları uzaya yayılarak parçacıklar (*particles*) üzerinden sahne temsil edilmeye çalışılır. Her parçacık yani 3B Gaussian, 3B uzayda konum bilgisine (*position*), yönelim bilgisine (*orientation*) ve tekdüze olmayan bir ölçeğe (*non-uniform scale*) sahiptir. Sahne temsili sonrasında parçacıklar 2B görüntü düzlemine normal dağılım kullanarak iz düşürülür ve bu düzensiz şekillendirme (*splatting*) olarak isimlendirilir. Buradaki düzensiz şekilden kasıt katı bir yüzeye çarpan viskozitesi yüksek yani ağdalı bir sıvının leke gibi görünen düzensiz şekil oluşturmasıdır. Bu yöntemde render işlemi parçacıkların splatting işlemidir yani 3B Gaussianların 2B uzayda oluşturulmasıdır (Kerbl vd., 2023). Özetle 3DGS, 3B noktaları 2B uzaya iz düşüren bir rasterlaştırma tekniğidir.



Şekil 11: *3DGS yönteminin çalışma şekli (Kerbl vd., 2023)*

3DGS yönteminde görüntülerin veya videolardaki çerçevelerin görüntü elde etme anındaki kamera konumları elde edilir. Görüntülerden nokta bulutları oluşturmak için Structure from Motion (SfM) yöntemi COLMAP sinir ağı kullanılarak oluşturulur. Her nokta normal dağılım özelliği gösterecek hale dönüştürülür. Nokta bulutlarını gerçek görüntülerle kıyaslayabilmek için önce nokta bulutları Şekil 11'de görüldüğü üzere diferansiyel Gauss rasterlaştırma tekniği kullanılarak rasterlaştırılır. Böylece gerçek görüntü ile nokta bulutları üzerinden farklılıkları hesaplayacak bir yitim (*loss*) fonksiyonu üretilebilir. Yitim değerlerine bağlı olarak normal dağılım parametreleri tahmin edilir. Normal dağılım parametreleri yani uzaydaki noktaların 3B Kartezyen koordinatları (X, Y, Z), 3x3 boyutundaki kovaryans matrisleri, noktaların renkleri (RGB) ve ışınım değerleri (σ) yapay öğrenme modeli ile hesaplanır. Eğitim aşamasında Stochastic Gradient Descent yöntemi kullanılır. Her bir piksel için parametreler hesaplanmaktadır. Daha sonra uzaya saçılan nokta bulutlarından elde edilen piksellerin gerçek görüntülerdekine belirli bir düzeyde benzetim sağlayana kadar konumları ve renkleri eşleştirilmeye çalışılır. Belirli bir düzeyde eşleşme sağlandığında eğitim süreci sonlandırılır (Kerbl vd., 2023).

4.4.3 Üç Boyutlu Temsildeki Gelişmeler

NeRF ile 3DGS teknikleri arasındaki temel fark sahne temsili için NeRF tekniğinde bir fonksiyon kullanılırken GS tekniğinde nokta bulutları parçacıklar (*3D Gaussian*) olarak kullanılmaktadır. Bu temel fark bir sahnenin gerçek zamanlı render işleminin gerçekleştirilmesinde 3DGS tekniğinin NeRF tekniğine göre daha verimli çalışmasını yani gerekli hesaplama maliyetinin daha düşük olmasını sağlamaktadır.

Burada önemli konu sahne temsilinde öne çıkan bu tekniklerin uygulamada nasıl kullanılacağı ve kullanımının nasıl yaygınlaştırılabileceğidir. Bunun için öncelikle veri boyutunun ve bellek kullanımının azaltılabilmesi gereklidir. Hesaplama maliyetinin azalması bu tür yöntemlerin akıllı telefonlar gibi düşük hesaplama/işlemci gücüne sahip cihazlarda gerçek zamanlı olarak kullanılabilmesinin önünü açmaktadır. Ayrıca başka uygulama alanlarında ve araçlarda kullanabilmek için Evrensel Işınım Formatı (*Universal Splat/Radiance Format*) gibi bir format oluşturulabilir. Örneğin oyun motorlarıyla, WegGL, WebXR, WebGPU, Evrensel Sahne Açıklaması (Universal Scene Description, USD) gibi web ve bilgisayar grafiği teknolojileri ile nasıl bütünleşebileceği üzerine çalışılabilir. Bunula birlikte büyük dil modellerindeki anlamsal yaklaşım (*semantic understanding*) sorunu SMERF tekniği içinde geçerlidir. Bu konudak gelişmelerle sahnedeki nesneler için semantik bölütleme (*semantic segmentation*) gerçekleştirilebilir.

NeRF alanında, eğitim sürecini hızlandırmak için grid tabanlı temsiller kullanılması sonucunda çentik (*aliasing*) türünde bozucu etki sorunları ortaya çıkmaktadır. Bu sorunu gidermek için kenar yumuşatma (*anti-aliasing*) yöntemi olarak ışık ışınları yerine koni geometrisini kullanan mip-NeRF 360 tekniği geliştirilmiştir. Ancak bu yaklaşım hızlandırılmış eğitim olanağı sunan Instant NGP (iNGP) gibi mevcut grid tabanlı tekniklerle uyumlu değildir. Bu kısıtı gidermek için Google tarafından Zip-NeRF adı verilen yeni bir model geliştirilmiştir. Zip-NeRF, hızlı eğitim ve kenar yumuşatma arasında bir ödünleşim (*trade-off*) çözümü sunmaktadır. Zip-NeRF sayesinde 3B sahnelerin yüksek kalitede modellenmesi ve render edilmesi için yeni olanaklar ortaya çıkmıştır (Barron vd., 2023). Öyle ki Google önce "Etkin Bellek Kullanımlı Işınım Alanlarını (Memory Efficient Radiance Fields, MERF)" ve daha sonra Zip-NeRF ile beraber "Aralıksız Olarak Aktarılabilen Etkin Bellek Kullanımlı Işınım Alanlarını (Streamable Memory Efficient Radiance Fields, SMERF)" geliştirmiştir. Her ne kadar MERF etkileyici bir gelişme olsa da büyük sahnelerde zorlanması nedeniyle MERF tekniğinden SMERF tekniğine geçiş önemli bir gelişme olarak kabul edilmektedir. SMERF ile akıllı telefonlar ve dizüstü bilgisayaralar gibi günlük kullanım cihazlarında 60 fps gibi olağanüstü bir hızda çalışan Zip-NeRF kalitesinde NeRF'ler gerçek zamanlı olarak elde edilebilmektedir.

SMERF hesaplama yükünü ve bellek tüketimini azaltmak için daha önce PyNeRF tekniğinde uygulanana benzer biçimde hiyerarşik model bölümlendirme (*hierarchical model partitioning*) kullanmaktadır. Böylece sahne daha küçük bölümlere (*segments*) ayrılmakta ve her bir bölüm farklı NeRF modeliyle temsil edilerek render işlemi optimize edilmeye çalışılmaktadır. Kullanıcı hangi bölümleri görmek isterse istediğinde o bölümler hızlı bir şekilde render edilmektedir.

SMERF tekniğinde; renk, doku gibi görsel görünüm parametrelerinin hesaplanmasını, şekil, derinlik gibi geometrik parametrelerin hesaplamasından ayırmaktadır. Bu da SMERF tekniğinin MERF tekniğine göre aynı hesaplama gücünü kullanarak daha yüksek model kapasitesi oluşturabilmesini sağlamaktadır.

SMERF tekniğinde öznitelik ayırma (*feature gating*) özelliği kullanılarak render işlemi için en uygun özniteliklere odaklanılması sağlanmaktadır. Böylece öğrenme ve render süreçleri optimize edilebilmektedir.

3DGS kadar hızlı ve ondan daha az bellek kullanarak daha iyi sonuçlar veren "Genelleştirilmiş Üstel Düzensiz Şekillendirme (Generalized Exponential Splatting, GES)" yöntemi bulunmaktadır (Hamdi vd., 2024).

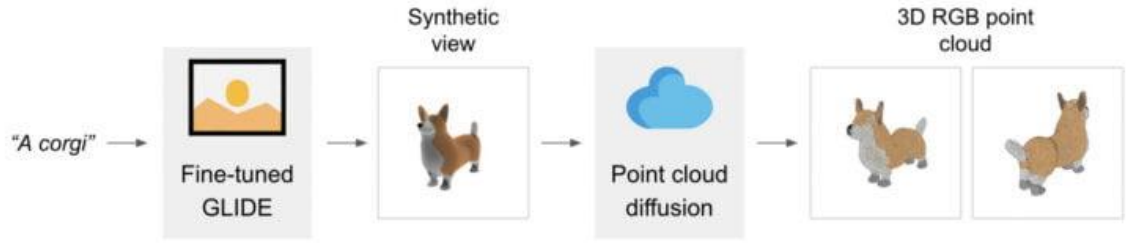
4.4.4 Üç Boyutlu Modelleme için Üretken Yapay Öğrenme

Işınım alanlarının yüksek kalitede gerçek zamanlı olarak render edilebilmesi, metinsel ifadeden 3B model (*text-to-3D generator, text-to-3Dmodeling*) üreten uygulamalar konusunda gelişmelere neden olmuştur. Böylece 3B model üretimi uygulamalar üzerinden herkes tarafından erişilebilir duruma gelmiştir. Bugün gelinen nokta bu yöntemlerin varyantlarını geliştirerek daha hızlı, daha iyi kalitede ve daha düşük hesaplama maliyeti ile 3B modellerin oluşturulmasıdır.

Bunun için difüzyon modelleri 3DGS ile beraber kullanılabilir. 3DGS ile nokta bulutu temelli 3B difüzyon modelleri oluşturulabilir, daha sonra geometriler ve görünümler 2B difüzyon modelleri ile iyileştirilebilir. OpenAI firmasının Point-E, Nvidia firmasının Magic3D, Google firmasının DreamFusion çözümleri benzer yaklaşımları kullanmaktadır.

Point-E uygulaması metinsel ifadeden 3B modeli değil, 3B şekli (*3D shape*) temsil eden 3B nokta bulutlarını üretmektedir (*text-to-3D generator*). Diğer bir ifadeyle 3B Gauss yapısındaki parçacıkların başlatılacağı bir nokta bulutu oluşturulmaktadır. Point-E uygulamasıyla ile bir nokta bulutu üretildikten sonra başka bir model bu nokta bulutunu mesh gibi bir yapıda (*text-to-mesh*) 3B modele dönüştürebilir (*transform*).

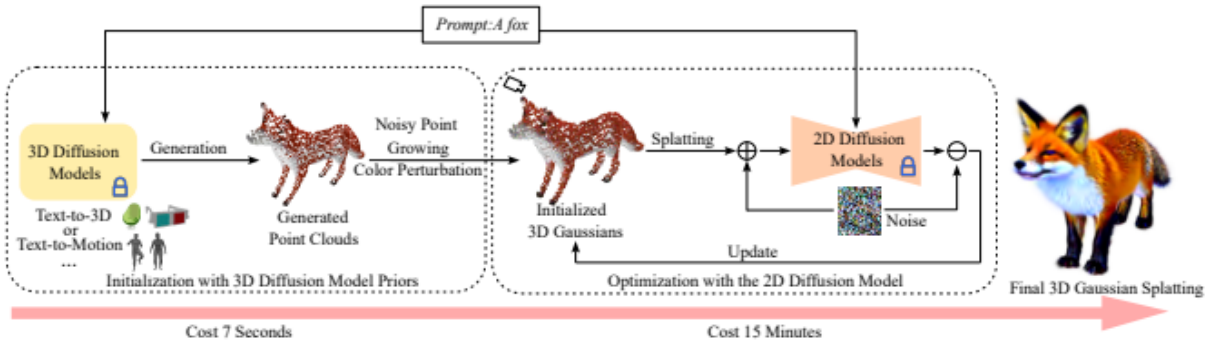
Point-E uygulamasında şekil 12'de görüldüğü üzere GLIDE ve görüntüden 3B model (*image-to-3D model*) olmak üzere iki üretken model bulunmaktadır. GLIDE, DALL-E uygulamasının metinsel ifadeden görüntü oluşturmasına benzer biçimde metinsel ifadeden görüntüler (*text-to-image*) üretmektedir. Diğer model OpenAI tarafından görüntüler ve ilişkili 3D nesneler üzerinde eğitilmiştir. Böylece görüntülerden 3B nokta bulutlarını çıkarılabilmektedir (Nichol vd., 2022).



Şekil 12: *Metinsel ifadeden 3B nokta bulutu üretiminin iş akışı (Nichol vd., 2022)*

DreamFusion uygulamasında 3DGS yerine NeRF yaklaşımı kullanılmıştır. Google firması Dreamfusion uygulamasında NeRF yönteminin 3B görüntü oluşturma yeteneklerini kendisinin önceden eğitilmiş metinsel ifadeden görüntü oluşturan difüzyon modeli olan Imagen ile bütünleştirmiştir. Dreamfusion 3B modeli, Imagen ile üretilmiş 2B sentetik görüntülerden üretebilmektedir. Nvidia firması Magic3D ürününde Instant Neural Graphics Primitives (Instant-NGP) çözümünü metinsel ifadeden görüntü oluşturan eDiffi çözümü ile bütünleştirmiştir. Farklı bakış açılarından eDiffi ile oluşturulan görüntüler 3B üretim için girdi (*input*) oluşturmaktadır. Daha sonra Nvidia'nın Instant NGP çatkısı bu görüntüleri 3B temsile dönüştürmektedir. Böylece Magic3D uygulaması metinsel ifadeden yüksek çözünürlüklü 3B nesneleri üretebilmektedir.

Şekil 13'te genel yapısı gösterilen Yi vd. tarafından geliştirilen GaussianDreamer ve Chen vd. tarafından geliştirilen GSGen3D çözümleri DreamFusion ve Magic3D uygulamalarından daha hızlı ve daha iyi ya da benzer sonuçlar üretebilmektedir (Chen vd., 2023; Yi vd., 2023).



Şekil 13: *GaussianDreamer çalışma şekli (Yi vd., 2023)*

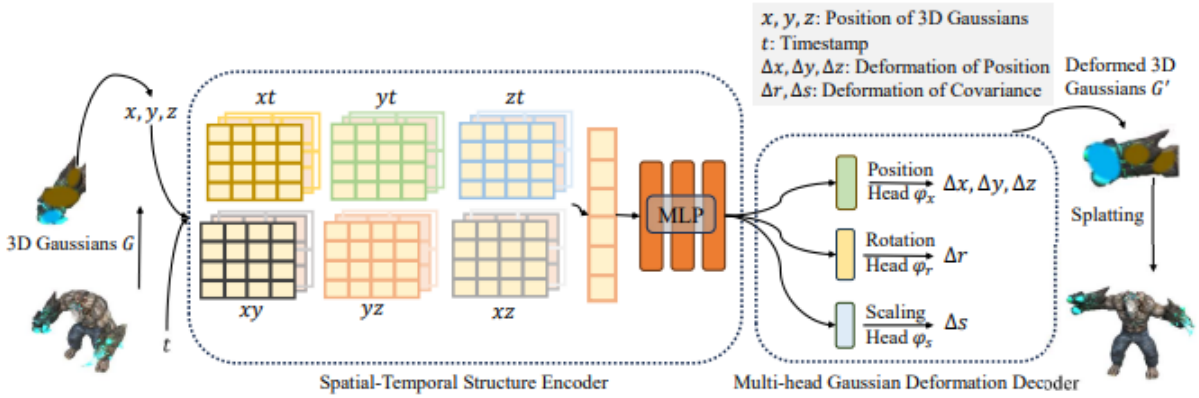
Görüldüğü üzere 3B eğitim verisi (*AI-ready data*) kısıtlı olduğu için firmalar 3B modeli oluşturmak için önce bir nesnenin farklı bakış açılarından 2B görüntülerini daha önceden geliştirdikleri üretken yapay zekâ uygulamaları ile oluşturmakta daha sonra bu 2B görüntüleri kullanarak 3B modeller oluşturmaktadır. Böylece son kullanıcı da dahil olmak üzere herkes tarafından doğal dil ile (yani metin tabanlı istemler girerek) 3B içerik, karmaşık bir sahne (*text-to-scene*) oluşturulabilmektedir. Metinsel ifadeden 3B sentetik görünüm (*text-to-3Dsynthesis*) tekniği ile 3B bir modelin önden-görünüm, üstten-görünüm gibi farklı açılardan görünümleri elde edilebilir. Hatta kullanıcı tarafından bakış yönüne bağlı istemler girilerek modeller üretilebilir. Bu tür uyugulamalarda üretilen 3B modeller mesh olarak dışarı aktarılabilmekte ve 3B modelleme yazılımlarında kullanılabilmektedir. Ayrıca Unity, Unreal Engine gibi oyun motorlarının da 3DGS, NeRF gibi render yöntemlerini kullanılabilir hale getireceği öngörülmektedir.

4.4.5 Dinamik Sahne ve Hareketli Nesneler

Ortamın zaman içinde hareket eden ve dönen bir dizi 3B Gauss özellikli parçacıklar olarak modellenmesi 3DGS yönteminden 4DGS yöntemine geçiş olarak kabul edilebilir. Bu da üretken yapay öğrenme modelleri için metinsel ifadeden 4. boyuta (*text-to-4D*) olan yeni bir üretim şeklidir. Metinsel ifadelerden GS4D ve difüzyon modeller kullanılarak 4B görünüm sentezi (*text-to-4DSynthesis*) oluşturulabilmektedir. Diğer bir ifadeyle hareketin sentezi (*motion synthesis*) gerçekleştirilebilmektedir.

Luiten vd. tarafından 2023 yılında yapılan çalışmada Dinamik 3B Gauss yaklaşımı kullanılarak dinamik sahne görünüm sentezini ve sahnede bulunan nesnelerin 6 serbestlik dereceli takip (*tracking*) görevini eş zamanlı olarak gerçekleştirebilen bir yöntem sunulmuştur. Çalışma kapsamında dinamik sahneleri modellemek için, 3B Gauss özellikli parçacıkların zaman içinde hareket etmesine ve dönmesine izin verilirken, kalıcı renk, ışık yoğunluğu ve boyuta sahip olmaları zorunlu kılınmıştır (Luiten vd., 2023).

Şekil 14'te çalışma şekli gösterilen diğer bir çalışmada hem parçacık hareketlerini hem de şekil deformasyonlarını modellemek için etkin bir deformasyon alanı oluşturulmuştur. Dinamik sahnelerin gerçek zamanlı render işlemlerinde yüksek görüntü çözünürlüklerinde render kalitesi korunabilmiştir (Wu G. vd., 2023).



Şekil 14: *Gerçek zamanlı dinamik render işleminin çalışma şekli (Wu G. vd., 2023)*

Yukarıdaki iki çalışma; otonom sürüşten gezgin robotiğe ve insansı robot alanına, oyun geliştirmeden endüstriyel uygulamalara kadar birçok alanda kamera görüntülerinden ortamdaki nesnelerin boyutunu ve şeklini daha iyi anlamak ve dinamik 3B ortamda nesneleri izlemek için yeni bir çözümün ortaya çıktığını göstermektedir.

5. Üretken Yapay Öğrenme Uygulaması Geliştirme

Kurum ve kuruluşların üretken yapay zekâ uygulamalarını ölçeklenebilir seviyede geliştirmek ve/veya kullanmak için bir mimari çözüme gereksinimleri bulunmaktadır. Üretken yapay öğrenme uygulamalarını çalıştırmanın temel olarak kendi kendine (*self-hosted*) veya hizmet olarak alma (Software-as-a-Service, SaaS) biçiminde iki farklı yolu bulunmaktadır.

Kapsamlı akıl yürütme becerileri gerekmeyen ve bilgilerin ayrıntı düzeyinin yönetilebilir olduğu uygulamalarda büyük dil modelleri verimli çalışmaktadır. Bununla birlikte, bilgi yoğun metinlerle veya karmaşık akıl yürütme gerektiren bağlamlarla uğraşırken ince ayar gerekmektedir. İnce ayar, büyük dil modellerine büyük miktarda bilgiyi sindirme, özetleme ve yorumlama gücü vererek daha etkili düşünmelerini ve akıl yürütmelerini sağlamaktadır. Böylece büyük dil modelleri iyileştirilmiş ve bağlamla ilgili yanıtlar verme yeteneğini geliştirilmiş olur.

Özelleştirilmiş çok modlu yapay öğrenme uygulamalarını geliştirmek için bir geliştirme çatkısı (*framework*) kullanılmak durumundadır. Böylece istemleri ve bilgi tabanlarını (*knowledgebases*) karmaşık uygulama programlama arayüzlerine (API) zincirleyerek karmaşık uygulamaları geliştirmek, düzenlemek ve yönetmek olanaklı hale gelmektedir. Bu tür geliştirme çatkılarına LangChain, AutoGen, FlowiseAI, Auto-GPT, AgentGPT, BabyAGI, LangDock, GradientJ, TensorFlow, LlamaIndex örnek olarak gösterilebilir.

Hugging Face, Github gibi kod alanlarında bulunan mevcut açık kaynaklı yapay öğrenme modelleri, GPT uygulama arayüzleri kullanılarak yapay zekâ temelli sohbet robotu, web tarayıcıları için yapay zekâ araçları gibi uygulamalar ve başka tür görevleri yerine getiren büyük dil modeli uygulama arayüzleri geliştirilebilmektedir.

Geliştirme çatkıları arasında en çok öne çıkan açık kaynaklı olan LangChain çatkısıdır. LangChain bir python çatkısıdır. LangChain çatkısındaki zincir (*chain*) özelliği bir modelin çıktısını başka bir modele girdi olarak verilmesini sağlar. Diğer bir ifadeyle farklı büyük dil modelleri birlikte kullanılabilir. LangChain çatkısında diğer bir özellik asenkron çalışabilen etmenlerdir. Etmenler büyük dil modellerinden gelen sonuçlara bağlı olarak eyleme geçebilir.

"Küçük Dil Modelleri (Small Language Model, SLM)", "Büyük Görsel Modeller (Vision Large Models, LVM)" gibi temel modelleri (*foundation models*) sunucu çiftlikleri yerine kişisel bilgisayarlarda, akıllı telefonlarda, Raspberry Pi gibi tek kart

bilgisayarlarında (Single-Board Computers, SBC) çalıştırmak için açık kaynak kodlu Ollama çatkısı kullanılabilir. Ollama ile küçük dil modelleri olarak Mistral, Phi-2 ve LVM olarak açık kaynak kodlu LLaVA ile üretken yapay zekâ uygulamaları geliştirilebilir. Ollama model çıkarımı için bir REST API olanağı da sunmaktadır. Böylece diğer uygulamalardan ve cihazlardan bu modele erişim sağlanabilmektedir.

Kurum ve kuruluşların sahip oldukları yapılandırılmış ve yapılandırılmamış veri kümeleriyle birlikte üretken yapay öğrenme modellerini kullanarak mevcut bulut altyapılarında analiz ve veri analitiği yapmak istediklerinde mikroservislerle büyük dil modellerini ölçekleyebilmeleri gerekmektedir. Bu da olay güdümlü (*event-driven*) hesaplama ile mikroservis mimarisinin büyük dil modelleriyle birlikte kullanılmasını gerektirmektedir. Kuruluşların kendi büyük dil modellerini geliştirmelerini ve kullanımlarını kolaylaştırmak için Large Language Model Operations (LLMOps) olarak bilinen yeni bir yöntem ortaya çıkmıştır. Bu yöntemle büyük dil modellerinin orkestrasyonu yapılabilmektedir.

6. Büyük Dil Modelleriyle Doğal Dil Sorgularından Yapılandırılmış Bilgi Çıktısı Üretme

Her ne kadar büyük dil modelleri metin üretme konusunda iyi kabul edilse de yanıtlarını yapılandırılmış veri olarak üretmemekte veya yapılandırılmamış veriyi XML, JSON/GeoJSON gibi çıktı formatlarında (*serialization formats*) sunmakta zorluklar yaşamakta ya da bu çıktı format yapılarını güvenilir şekilde oluşturamamaktadır. Çünkü JSON gibi formatlar söz dizimsel (*syntactically*) olarak doğru olmalı ve yapısını belirten bir şemaya uygun olmalıdır. Öyle ki büyük dil modeline geçerli (*valid*) bir JSON olarak yapılandırılmış bir yanıt üretmesi için JSON yapısı/şeması verilse bile büyük dil modeli uygulaması kullanıcının talimatlarına esnek bir şekilde bağlı kalmakta ve kullanıcının belirtmediği bazı anahtarları (*keys*) uydurabilmektedir. Ayrıca bazı durumlarda JSON çıktısı doğru şekilde ayrıştırılamamaktadır (*parsing*). Hatta çıktı ayrıştırıcısı olarak (*output parser*) LangChain Pydantic Output Parser bile kullanılsa istemde belirtilen JSON şemasının her zaman başarılı çalışacağının garantisi bulunmamaktadır. Bu durum büyük dil modellerinin çıktılarını başka sistemlerde/uygulamalarda girdi olarak kullanmayı zorlaştırmaktadır. Özetle büyük dil modelleriyle doğal dil sorgularından yapılandırılmış bilgi çıktısı üretme konusunda aşağıda ifade edilen sorunlar bulunmaktadır:

- Büyük Dil Modelleriyle doğal dil sorgularından doğru, güvenilir ve temiz JSON/GeoJSON çıktısını üretme.
- Büyük Dil Modelleriyle JSON verisi üzerinde doğal dil sorgusu gerçekleştirme ve yanıtları insan tarafından okunabilir bir formatta alabilme.

Bu sorunlara ilişkin OpenAI şirketi kendi ekosistemine yönelik olarak Haziran 2023 tarihinde "*function calling*" özelliğini eklemiştir. Ancak bu çözüm yalnız OpenAI ekosisteminde, OpenAI API ile birlikte kullanılabilmektedir. Birlikte çalışabilirlik ilkesi kapsamında yani diğer üretken modellerle ve sağlayıcılarla birlikte kullanım açısından "*function calling*" uygun bir çözüm değildir. Instructor kütüphanesinde doğrulanmış (*validated*) ve okunabilir (*readabale*) JSON çıktıları üretmek için OpenAI function calling özelliği LangChain Pydantic doğrulayıcılarıyla (*validators*) bütünleştirilmiştir.

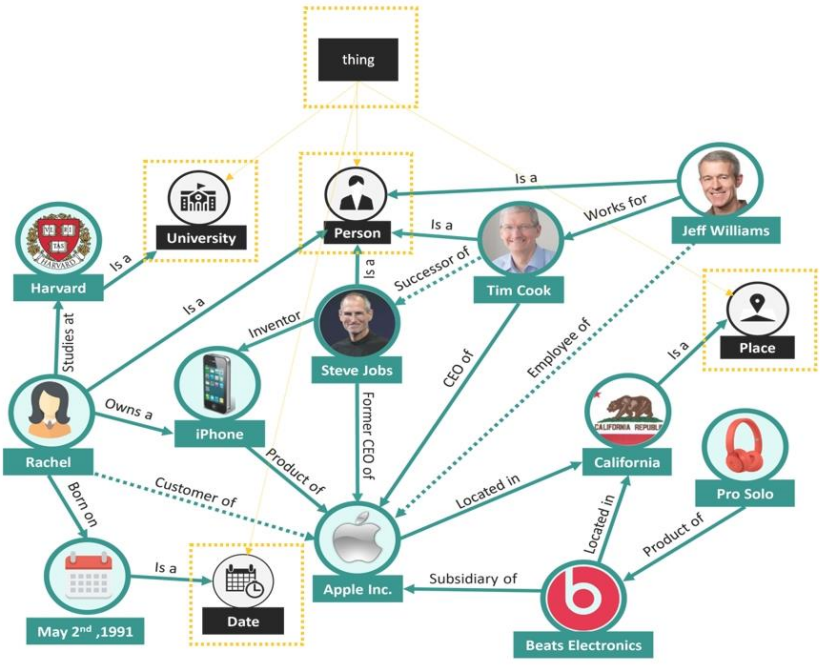
Alternatif olarak açık kaynak kodlu jsonformer kütüphanesi kullanılabilir. Jsonformer, Hugging Face transformers kütüphanesi ile entegre edilebilir. Ancak Jsonformer, önceden tanımlanmış JSON şemalarının kullanımını gerektirir. Diğer bir ifadeyle bu çözümle büyük dil modeli dinamik olarak şema oluşturamamaktadır. Ayrıca OpenAI veya diğer API'ler gibi Hugging Face kod alanında olmayan modellerle birlikte çalışamamaktadır. Jsonformer aktif olarak gelişim gösteren bir kütüphane değildir. Diğer bir alternatif de outlines kütüphanesidir. Bu çözümde sorunsuz bir şekilde LangChain Pydantic modeline uygun JSON üretilebilir. Aktif olarak geliştirilmekte olan bu kütüphane, yalnız JSON yanıtları için değil, diğer yapılandırılmış metin oluşturma kullanım durumları için de en gelişmiş çözümü sunmaktadır.

Başka bir yaklaşım da Dil Modeli Sorgulama Dili (Language Model Query Language, LMQL) olup büyük dil modelleri ile makine öğrenmesinde standart betimleme dili olan Python dilini birleştirmektedir (Beurer-Kellner vd., 2023).

7. Bilginin Temsili ve Gelişkin Akıl Yürütme

Her ne kadar bugün mevcut içeriklerin çoğu yapılandırılmamış veri türünde olsa da kamu kurumları ve özel sektör kuruluşları çoğunlukla veri tabanlarında bulunan yapılandırılmış veri (*structured data*) kümeleri üzerinden çalışmaktadır. Kurum ve kuruluşların büyük dil modelleri temelli üretken yapay öğrenme modellerini kendi iş süreçleri ile bütünleştirebilmeleri için büyük dil modelleri ile yapılandırılmış veri kümelerini daha da önemlisi bilgi çizgelerini birlikte kullanabilmeleri gerekmektedir.

Bilgi çizgesi (*knowledge graph*), birbiriyle bağlantılı varlıklardan oluşan bir ağdaki bilgileri temsil eden (*representation*) gelişmiş bir veri yapısıdır. Bir bilgi çizgesi, alana özgü anlamların düğümler (*nodes*) ve kenarlarla (*edges*) ilişkilendirildiği etiketli ve yönlendirilmiş bir çizgedir. Sözü edilen çizge birbirine bağlı varlıklar (*entities*) arasındaki semantik bilgiyi temsil eder. Yapılandırılmış veri kümelerini içeren bilgi çizgeleri Şekil 15'te görüldüğü gibi birbirinden farklı ve ilgisiz görünen bilgiler arasındaki karmaşık ilişkileri ortaya çıkarmak için geliştirilmiştir. Şekil 15'te noktalı dikdörtgenler yer, tarih gibi soyut kavramları, elipsler yani düğümler California, Mayıs 1991 gibi somut kavramları, oklar yani kenarlar kavramlar arasındaki ilişkileri, oklar üzerindeki yazılar etiketleri yani öznitelikleri/ilişkileri göstermektedir. Bilge çizgeleri; veri kümelerini ve ilişkilerini hiyerarşik olarak makine tarafından anlaşılabilir bir formatta düzenlemektedir. Böylece makinelerin insanın düşünme şekline göre varlıkların birbirleriyle nasıl ilişkili olduğunu ve paylaşılan özniteliklerini (*attributes*) bağlam içinde anlayabilmelerini sağlar. Bilgi çizgesindeki bilgiler genellikle bir çizge veri tabanında (*graph database*) saklanır ve Şekil 15'teki gibi bir çizge yapısı (*graph structure*) olarak gösterilir. Semantik bilgi çizgileri aynı zamanda semantik web teknolojisi olarak da kullanılmaktadır. Bu tür semantik web teknolojilerine örnek olarak DBPedia, GeoNames, Wordnet, Google Knowledge Graph, FactForge, Wikidata, Linked Open Data Cloud örnek olarak gösterilebilir. Özellikle açık bağlantılı veri (*Open Linked Data*) hareketi bu konudaki gelişmelere yön vermektedir. Ontolojiler kullanılarak hem insanlar hem de makineler tarafından tümdengelimli akıl yürütme (*deductive reasoning*) ve çıkarım yapılabilir.



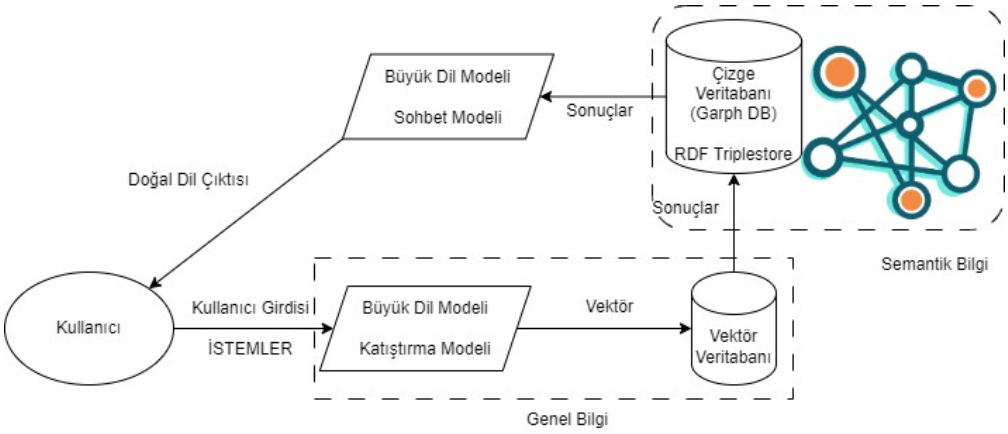
Şekil 15: *Bilgi çizgesi gösterim örneği (URL-2)*

Birbirinden farklı ancak birbirini bütünleyen iki ayrı çalışma alanı olan yapılandırılmış veriye dayalı bilgi çizgeleri ile yapılandırılmamış veriye dayalı olan büyük dil modelleri arasındaki simbiyotik ilişki kurulup sürdürülebilirse büyük dil modelleri dolayısıyla üretken yapay öğrenme modelleri daha etkin biçimde kullanılabilir. Büyük dil modellerinin metinsel ifadeyi anlama ve üretme yetenekleri, bilgi çizgelerinin anlamsal zenginliği, yapılandırılmış veri temsili ve olgusal (neden-sonuç ilişki kurabilme) üstünlükleri ile uyumlu hale getirildiğinde üretken yapay öğrenme modellerinin akıl yürütme ve çıkarım yapma yetenekleri daha da gelişecek ve böylece bağlam farkındalığı yüksek, daha doğru, güvenilir ve açıklanabilir/yorumlanabilir çıktılar elde edilebilecektir.

Büyük dil modelleri (vektör veri tabanları) ve bilgi çizgeleri (çizgeler ve çizge veri tabanları) karşılıklı olarak birbirlerini geliştirme potansiyeline sahip birbirini bütünleyici teknikler olduğu Pan vd. tarafından detaylı olarak incelenmiş ve aşağıda özetlenmiştir:

- Bilgi çizgeleri ile geliştirilen büyük dil modelleri diğer bir ifadeyle büyük dil modellerinin bilgi çizgeleri üzerinde eğitilmesi (*full training*); üretken yapay öğrenme modellerinin bağlamsal ve olgusal bilgileri daha iyi anlamasını sağlayarak çıktı kalitesini artıracaktır (Pan vd., 2024).
- Bilgi çizgeleri ile büyük dil modelleri eğitilmeyip yalnız ince ayar (*fine-tuning*) yapılması bir tematik alana özgü yapısal bilginin eklenmesini sağlayacak, çıktıların yorumlanabilirliği artacak ve daha bilgiye dayalı yanıtlar elde edilebilecektir (Pan vd., 2024).

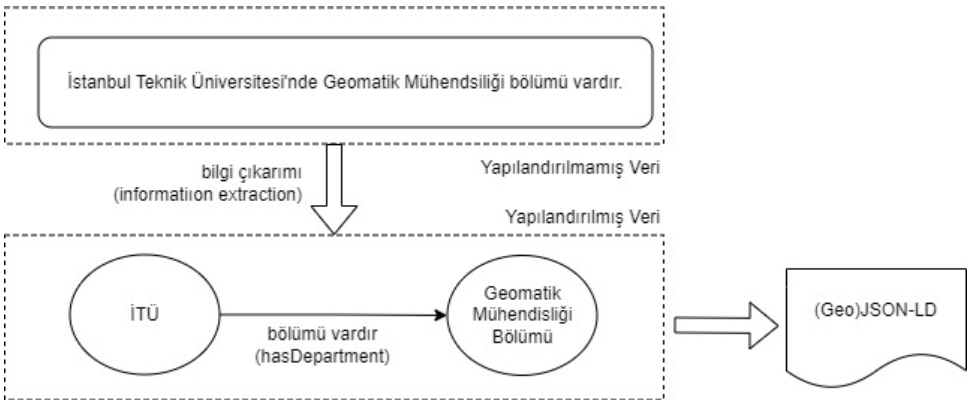
Şekil 16 çalışma kapsamında bu iki yaklaşımın nasıl birlikte kullanılabileceğine ilişkin olarak tasarlanmıştır. Şekil 16'da görüldüğü üzere ilk olarak büyük dil modeli vektör arama kullanarak ilgili bağlamsal bilgilere ulaşır. Ardından semantik arama ile yanıt zenginleştirilir.



Şekil 16: *Büyük dil modellerinin etkinliğini arttırmak için bilgi çizgelerinden yararlanma*

Bilgi çizgeleri, büyük dil modelleri için "Retrieval Augmented Generation (RAG)" görevi görebilir. Böylece, büyük dil modellerinin halüsinasyonları önlenirken daha kesin, doğru ve bağlamsal olarak ilgili çıktılar üretilebilecektir. Graph RAG, Vector RAG, Text2Cypher gibi farklı RAG teknikleri bulunmaktadır. Graph RAG bilgi çizgelerine dayalı bir erişim geliştirme tekniğidir ve vektör veritabanlarıyla, LangChain çatkısıyla birlikte çalışabilmektedir. LangChain, çizgenin şemasını okuyan ve kullanıcı girdisine göre uygun Cypher ifadelerini (*statements*) oluşturan bir GraphCypherQAChain özelliğine sahiptir.

Büyük dil modelleri bilgi çıkarımı (*information extraction*) için de kullanılabilir. Bilgi çıkarımının amacı Şekil 17'de görüldüğü üzere yapılandırılmamış metinden yapılandırılmış bilgiyi elde etmektir. Bilgi çıkarımı aşağıdaki şekilde görüldüğü üzere bilginin çizge temsiliyle (*graph representation*) sonuçlanmaktadır.



Şekil 17: *Yapılandırılmamış metinden yapılandırılmış bilgiye dönüşüm (transformation)*

Yapılandırılmamış metinsel ifade büyük dil modelleriyle önceden tanımlanmış bir ontolojiye (*pre-defined ontology*) uygun bir bilgi çizgesine dönüştürülebilir (*transform*). Diğer bir ifadeyle metinsel ifadenin temsili çizge yapısına yani bilgi çizgesine dönüştürülebilir (*text-to-graph*). Bunun için büyük dil modelinin hangi ontolojiyi/şemayı kullanacağına karar vermesi gerekmektedir. Çünkü ardından oluşacak "Yapılandırılmış Sorgu Dili (Structured Query Language, SPARQL)", GeoSPARQL, PathQL, GraphQL-LD, Cypher gibi sorgulara büyük dil modelinin yanıt verebilmesi için şema bilgisi gerekmektedir. SCHEMA.ORG, FOAF, SKOS, RDF, RDFS, OWL, JSON-LD, GeoJSON-LD vb. çeşitli standart ontolojiler üzerinde önceden eğitilmiş büyük dil modelleri kullanılabilir. Bu tür bir dil modeli ile kullanıcı hangi ontolojiyi kullanmak istediği bilgisini, bilgi çizgesine dönüştürmek istediği yapılandırılmamış bilgiyle birlikte isteminde belirtmelidir. Farklı dokümanlara ilişkin bilgi çizgesi oluştururken uygulamanın tek bir istemde izin verdiği token sayısı içerisinde işlem tamamlanamayabilir. Bu durumda farklı istemlerde oluşturulacak bilgi çizgelerinin aynı ontolojiye uygun olmasına dikkat edilmelidir.

Metinsel ifadeden çizgeye dönüşüm büyük dil modelinin önceden eğitildiği standart ontolojilerle sınırlıdır. Standart olmayan veya özel bir ontoloji (*custom ontology*) kullanılmak istenildiğinde; kullanılmak istenen özel ontolojinin tamamı istem içerisinde ifade edilmesi gerekmektedir.

Önceden eğitilmiş standart ontolojilerle sınırlı olmamak ya da özel ontolojiyi isteme dâhil ederken token ek yüküne sahip olmamak için ontoloji ile ince ayarlanmış büyük dil modeli yaklaşımı kullanılabilir. Ayrıca bazı durumlarda birden fazla ontoloji kullanmak gerekebilir. Bu tür durumlarda yani standart ontolojilerle önceden eğitilmiş dil modellerine başka bir ontoloji ile ince ayar yapıldığında üretken yapay öğrenme modelleri ontolojiler arasında dönüşümler de yapabilir.

Konuya diğer yönden bakıldığında yani büyük dil modellerinin bilge çizgelerinin yeteneklerini arttırması açısından bakıldığında bilgi çizgelerinin akıl yürütme ve çıkarım yapma başarımları yükseltilebilir. Bilgi çizgelerindeki geleneksel yöntemler genellikle eksik bilgilerin olduğu durumlarda ve çizge oluşturmak için metin veri kümelerinin işlenmesinde sıkıntılar yaşamaktadır. Diğer taraftan sıkıntı yaşanan bu alanlar büyük dil modellerinin üstün olduğu alanlar olduğu için büyük dil modellerinin desteği ile bu zorluklar giderilebilir. Ayrıca büyük dil modelleri bilgi çizgelerinden bilgi elde edilmesini (*information retrieval*) kolaylaştırabilir. Geleneksel yaklaşımla bir programlama dili ile veri tabanı üzerinden arama yapmadan, büyük dil modelleri ile bilge çizgelerine doğrudan soru sorup yanıt alınabilir. Böylece büyük dil modelleri sayesinde bilgi çizgelerine erişimi ve kullanımı daha kolay hale gelecektir.

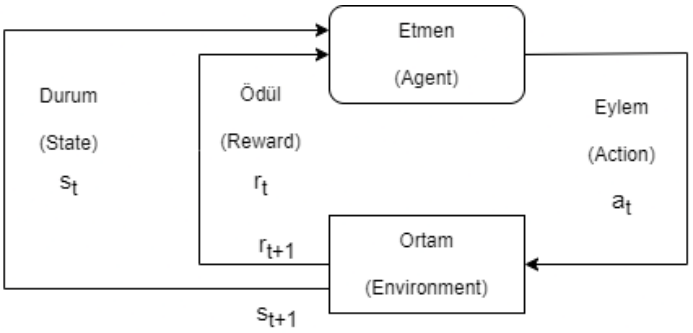
Görüldüğü üzere bilge çizgeleri ile büyük dil modelleri arasında bir sinerji bulunmaktadır. Her iki yaklaşımın birlikte kullanılması çift yönlü bir akıl yürütme ve çıkarım sağlayacaktır. Şekil 16'da gösterildiği gibi varlıklar ve ilişkileri göstermek için bir bilgi çizgesi kullanır ve ardından erişim geliştirme için büyük dil modeli kullanır. Böylece sorgular için daha kapsamlı

bir bağlamsal anlayış sağlanmış olur. Bunun sonucunda arama yapan kişinin beklentilerini daha iyi karşılayan, arama sonuçlarını daha düşük maliyetle elde edebilen uygulamalar ortaya çıkacaktır.

8. Planlama Algoritmaları

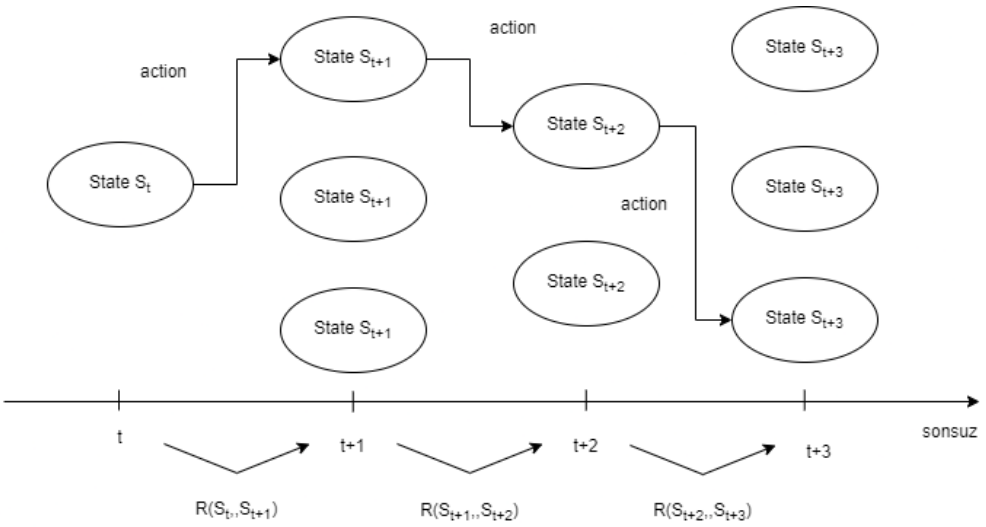
2016 yılında Google Deepmind firmasının AlphaGo modeli dünya Go şampiyonunu yenmiştir. 2018 yılında AlphaZero dokuz saat eğitimle en iyi satranç oyuncusu olmuştur ve o zamana kadarki en iyi satranç motoru olan Stockfish 8 uygulamasını yenmiştir. Ayrıca Shogi oyununda en iyi motor olan Elmo'yu da yenmiştir. 2019 yılında AlphaStar gerçek zamanlı bir strateji oyunu olan StarCraft II'de grandmaster seviyesine yükselmiştir. Üretken yapay zekâdan farklı olarak planlama yapabilen bu tür yapay öğrenme algoritmalarının başarısı yapay zekâ devrimi olarak görülebilir.

İnsan düşüncesinin makinedeki en iyi yansımaları satranç, Go, Shogi, StarCraft gibi oyunlarda gözlemlenmiştir. Bu oyunlarda karar verme veya problem çözme "Pekiştirmeli Öğrenme (Reinforcement Learning, RL)" üzerinden gerçekleşmektedir. Pekiştirmeli Öğrenme bir dizi karar (*sequence of decisions*) vermek için yapay öğrenme modellerinin eğitilmesidir. Şekil 18'de görüldüğü üzere pekiştirmeli öğrenmede belirsiz bir ortam (*uncertain environment*), bu ortamı gözleyen ve davranışlarını da buna göre değiştiren etmenler/karar vericiler (*agents/decision makers*) bulunur. Bu tür öğrenmede keşfedilecek belirsiz ortamla etkileşime giren etmenlerin ödülleri en üst düzeye çıkaracak (*maximize rewards*) eylemleri/hareketleri (*actions*) gerçekleştirmesi için bir eğitim (*train*) uygulanır. Bu eylemler sonucunda etmenin durumu (*state*) değişir, durum değişimi ile ödüller (*reward*) kazanır. Etmenin amacı ödülü en iyilemektir (*maximization*).



Şekil 18: *Pekiştirmeli öğrenmenin çalışma şekli*

Etmenin ortamla etkileşimleri zaman içerisinde sırayla (*sequentially*) gerçekleşir. Etmen her zaman adımında gerçekleştirdiği eylemle ortamın yeni temsiline yani yeni durumuna geçiş yapmaktadır. Eylemlerin sonucunda etmen ödül veya ceza almaktadır. Diğer bir ifadeyle etmen doğru eylemeleri/hamleleri yaptığında ödül kazanır yani eylemler olumlu sonuç vermektedir. Aksi durumda negatif ödül/ceza almaktadır. Belirli bir durumdayken bir eylem seçme, yeni bir duruma geçme ve bir ödül alma süreci sırayla tekrar tekrar gerçekleşmektedir. Etmenin amacı süreç boyunca ödül almak ve aldığı ödülleri en üst düzeye çıkarmak olduğundan etmenin yalnız anlık ödüller için değil süreç içerisinde alacağı kümülatif ödüller için en doğru eylemleri belirlemesi gerekmektedir. Eylemleri belirleyen algoritmaya politika (*policy*) denir. Pekiştirmeli öğrenmede belirsiz ortamın keşfedilmesi sırasında en iyi politika belirlenmeye çalışılmaktadır. En iyi politikayı belirlemek için Şekil 19'daki akışı yansıtan Eşitlik 1'in çözümü gerekmektedir. Bu eşitliğin çözüm süreci politika araması (*policy search*) olarak adlandırılır. Politika aramasında öne çıkan yöntem değer fonksiyon yöntemi (value-function method) olup bu yöntemde en bilinen algoritmalar Q-Öğrenme (*Q-Learning*) ve SARSA (*State-Action-Reward-State-Action*)'dır. Özetle, ortam politikayı belirler, politika da eylemeleri yani etmenin nasıl hareket edeceğini belirler. Böylece deneme yanılma yöntemi ile bir karara ulaşılır.



Şekil 19: *Politika belirleme*

Eşitlik 1'de $P(S) = A$ yani etmen/karar verici politika fonksiyonu (P) ile mevcut durumu (S) alır ve eylemi (A) üretir. $R_{P(S)}$ politika fonksiyonunun S durumu için olan ödülü karakterize eder. Politika fonksiyonu kümülatif ödüller için en üst düzeye getirilmeye çalışılır. Bu nedenle başlangıçtan sonsuza kadar olan bütün hamlelere/eylemlere ilişkin tüm ödüller toplam fonksiyonu ile bir araya getirilir. Gelecekteki hamlelere ilişkin ödüllerden emin olunmadığı için δ indirgeme faktörü ($0 < \delta < 1$) uygulanır.

$$\sum_{t=0}^{\infty} \delta^t \, R_{P(S)}(S_t, S_{t+1}) \tag{1}$$

Bir karar probleminde bir durumdaki değeri hesaplamak için Eşitlik 2'deki Bellman Denklemi (*Bellman Equation*) kullanılır. Bellman denklemi, belirli bir eylem seçiminin ne kadar iyi olduğunu açıklar. Eşitlik 2'de t anındaki durumun değeri $V(S_t)$ ile bir sonraki durumun değeri $V(S_{t+1})$ ile gösterilmiştir. $R(A,S)$ eylem gerçekleştirildiğinde alınacak ödülü ifade eder.

$$V(S_t) = max_A(R(A,S_t) + \delta \, V(S_{t+1}) \tag{2}$$

Bellman denklemini stokastik hale getirmek için Eşitlik 3'teki Markov Karar Süreci kullanılır. Markov Karar Süreçleri sıralı karar verme (*sequential decision making*) için olasılık tabanlı bir modeldir ve durum geçişlerinde (*state transition*) kullanılır. Markov Karar Süreçleri her adımda hangi eylemin gerçekleştirileceğine dair bir seçim yapılmasını gerçekleştiren bir süreçtir. Her eylem seçimi yeni durumları ortaya çıkarmakta ve böylece farklı durumlar için farklı olasılıklar ortaya çıkmaktadır. Markov'a göre gelecekteki durumun olasılığı, geçmişe değil, içinde bulunulan duruma yani mevcut duruma (*current state*) bağlıdır ve buna Markov Özelliği (*Markov Property*) denir.

$$V(S_t) = max_A(R(A,S_t) + \delta \sum_{S_{t+1}} P(S_t, A, S_{t+1}) V(S_{t+1}) \tag{3}$$

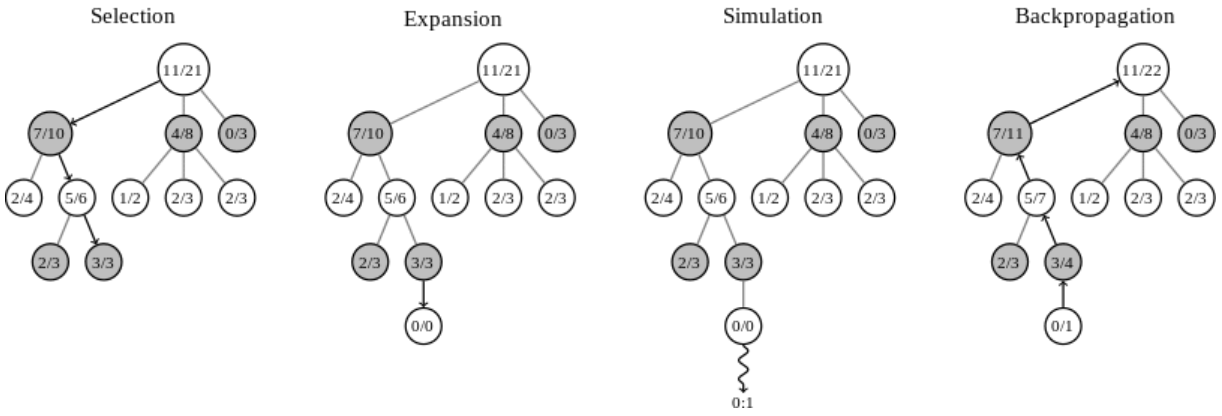
Pekiştirmeli öğrenme problemleri Markov Karar Süreci olarak modellenebilir. Pekiştirmeli öğrenme yöntemlerinden olan Q-Öğrenme her iki yaklaşımın bütünleştirilmesiyle ortaya çıkarılır. Bellman denkleminde ve Markov Karar Sürecinde durumun değeri $V(S)$ hesaplanırken, Q-Öğrenme yaklaşımında durumun eyleme bağlı kalitesi $Q(S,A)$ yani S durumunda A eylemi seçiminin ne kadar iyi olacağını hesaplanmaktadır. Q-Öğrenme yaklaşımında durumun değerinin hesaplanması yerine hangi eylemin daha iyi olduğu Eşitlik 4'teki Q-Öğrenme fonksiyonu ile belirlenir.

$$Q_{t+1}(S,A) = (1-\propto) Q_t(S,A) + \propto (R(S,A) + \delta max_{A'}(Q(S',A')) \tag{4}$$

Büyük dil modelleri temelli üretken yapay zekâ uygulamalarının akıl yürütmede diğer bir ifadeyle insan gibi düşünmede yaşadığı eksiklikleri tamamlamak için "Monte Carlo Ağaç Araması (Monte Carlo Tree Search, MCTS)" tekniğinden yararlanılır. Monte Carlo Ağaç Araması, "Markov Karar Süreci (Markov Decision Process, MDP)" için bir karar ağacı tarafından yapılandırılmış rastlantısal bölüm örneklemesine dayanan politika optimizasyon algoritmasıdır. Diğer bir ifadeyle MCTS, yapay zekâ problemlerinde optimal kararlar vermek için klasik ağaç arama algoritması ile pekiştirmeli öğrenmeyi bütünleştiren ileriye yönelik bir planlama algoritmasıdır.

MCTS, genellikle bilgisayar oyunlarında nihai kazanan çözüme ulaşmak için politikanın izlemesi gereken yolu/hamleleri tahmin etmek için kullanılır. Bir oyun ağacı, düğümleri (*nodes*) oyundaki konumlara (*positions*) karşılık gelen yönlendirilmiş bir çizgedir (*directed graph*). Bir yapay zekâ, oyun durumu (*game state*) olarak adlandırılan oyundaki mevcut konuma göre mümkün olan en iyi/uygun hamleyi seçmek, yani en iyi kararı vermek (*optimal decision*) için oyun ağacında arama yapar. Satrançta 10^{123}, Go oyununda 10^{360} ağaç karmaşıklığı yani potansiyel hamle ve karşı hamle sayıları varken ileriye yönelik planlamanın, oyun durumunu tahmin eden bir değerlendirme fonksiyonu ile yapılması kolay değildir. MCTS yaklaşımında insanlar tarafından yazılmış karmaşık sezgisel yöntemlere gereksinim yoktur. MCTS tekniğinde kullanan yapay öğrenme modelleri, programcıların herhangi bir doğrudan yönlendirmesi olmadan bir görevde nasıl ustalaşacaklarını öğrenebilirler. Bu özellik aynı zamanda MCTS tekniğinin uygulama alanından bağımsız (*domain-independent*) olmasını sağlamaktadır.

MCTS tekniği Şekil 20'de görüldüğü üzere seçim (*selection*), genişletme (*expansion*), benzetim (*simulation*) ve geri yayılma (*backpropagation*) olmak üzere dört temel adımdan oluşmaktadır. Düğüm seçimi aşamasında en umut verici düğüm (*parent*) seçilir. Genişletme aşamasında seçilen düğümden çocuklar (*child*) oluşturulur ve düğüm seçimi çocuklara doğru ilerler. Benzetim aşamasında en umut vaat eden çocuk üzerinden rastlantısal hamleler yapılır diğer bir ifadeyle oyun politikası belirlenir. Geri yayılım aşamasında yeni eklenen düğümle birlikte önceki tüm düğümlerin değerleri güncellenir. Düğümler durumların (*state*) temsilidir. Farklı düğüm seçimi farklı bir hamle yani farklı bir karar demektir. Her yapılan hamle veya verilen karar başka bir durumun ortaya çıkmasına neden olur. Bu adımlar en umut verici hamle bulunana kadar tekrarlanır.



Şekil 20: *MCTS tekniğinin ana bölümleri*

Etmen belirsiz ortamda rastlantısal olarak hareket ettiğinde ortamı keşfeder (*exploration*) yani global arama işlevini yerine getirir. Ancak bu durumda hedefe giden yolu tam olarak bulamaz. Diğer taraftan etmen keşfetmeden yalnız belirli aralıktaki değerler üzerinden hareket ederse (*exploitation*) yani o ana kadar bulunan en iyi stratejileri kullanırsa (lokal arama) bu kez de bulduğu yolun belirsiz ortamdaki en iyi yol olup olmadığını anlayamayacaktır. Politika olarak adlandırılan kavram bu global arama ile lokal arama arasındaki ödünleşimi sağlamaktadır (*exploitation-exploration trade off*) (Coulom, 2007).

9. Sonuçlar ve Öneriler

El-Cezeri ile birlikte 13. yüzyıldan itibaren makinelerin fiziksel gücü üzerine çalışılmaya, mekanik yapıların insan davranışlarını taklit edebileceği üzerine düşünülmeye başlanmıştır. 1950'li yıllardan itibaren insansı mekanik yapılardan insan gibi düşünebilen makineler kavramı tartışılmaya başlanmıştır. 1950 yılında Alan Turing "Computing Machinery and Intelligence" makalesinde "Bilgisayarlar Düşünebilir mi?" sorusunu sormuştur. 1959 yılında Ord. Prof. Dr. Cahit Arf bir konferansta "Makine Düşünebilir mi ve Nasıl Düşünebilir?" başlıklı bir konuşma yapmıştır. Bugün makinelerin düşünmesi akıl yürütme ve çıkarım yapma üzerinden gerçekleşmektedir. Hoffmann vd. tarafından önerilen Chinchilla ölçekleme yasasına göre yapay sinir ağındaki parametre sayısının artması modelin doğruluğunu artması ile yakından ilişkili olduğunu göstermiştir (Hoffmann vd., 2022). Buna göre işlenen veri arttıkça, hesaplama birimleri geliştikçe, bilgi işlem altyapısı kuvvetlendikçe, büyük dil modellerinin parametre sayısı arttıkça makineler çok daha iyi düşünebilecektir.

Diğer taraftan insanın düşünme sistemi; akıl/mantık yürütebilme, çıkarım yapabilme, öğrenme yeteneğine sahip olma, geçmişi hatırlayabilme, yaratıcı düşünebilme vb. görevlerden oluşur. Makinelerin, yapay zekânın öğrenmek için kullandığı veri/bilgi insanın yani doğal zekânın kültür ve bilgi birikiminden gelmektedir. Bu nedenle daha önce hiç yapılmamış işlerde ve sezgisel yaklaşım gerektiren konularda yapay zekâ başarılı olamamaktadır. Mevcut hali ile yapay zekâ iyi bir ezbercidir ve örüntüleri bulmada çok iyidir ancak yaratıcılıktan, sezgiden ve soyut düşünmeden hâlâ yoksundur.

Düşünsel sistemleri anlama ve yorumlama için geliştirilmiş olan ikili süreç teorisine (*dual process theory*) göre düşünme sistemi genel olarak otomatik süreçler ve bilinçli süreçler olmak üzere iki sınıfa ayrılmaktadır. Büyük dil modellerinin mevcut durumu daha basit düşünme yaklaşımı gerektiren otomatik süreçleri gerçekleştirebilirken, planlama algoritmaları daha sezgisel olan bilinçli süreçleri gerçekleştirebilmektedir. Örneğin büyük dil modelleri kendisine yöneltilen bir soruyu yanıtlayabilirken, otonom araç kullanımını gerçekleştirememektedir. Otonom araçta satranç, Go, StarCraft gibi oyunları oynayamamaktadır. Evi temizleyebilen robot süpürge metinsel ifadeden video oluşturamamaktadır. Oysaki farklı kuruluşlar ve özgün geliştiriciler tarafından geliştirilmiş farklı görevlerde özelleşmiş, uzmanlaşmış uygulamalar bulunmaktadır. Bu uygulamalar birlikte çalışabilir hâle geldiklerinde, bilgi çizgeleriyle eğitilebilir olduklarında, eylem modelleri ve planlama algoritmaları ile bütünleştiklerinde farklı düşünebilen, yaratıcı özellikler sergileyen yapay zekâdan bahsedilmeye başlanabilir.

Yapay öğrenme modellerinin insandan daha hızlı biçimde veri kümelerini işleyebilmesi, analiz edebilmesi ve buna bağlı olarak daha hızlı karar alması olağan olarak kabul görmektedir. Buna ek olarak insanın düşünmesini ve üretim süreçlerini tamamlayan bir asistan (ek zekâ) olarak görülmesi yerleşiklik kazanmaya başlamıştır. Bu nedenle yapay zekânın yalnız tabana yayılarak olabildiğince insanın kullanmasını sağlamanın yanında güvenli ve güvenilir yapay öğrenme modellerinin (*trusted AI*) iş birliği içinde açık ve açıklanabilir yapay zekâ olarak geliştirilmesi konusu önemlidir. Dev işletmelerin sermayelerini ve hesaplama kaynaklarını kullanarak geliştirip hâkim olmaya çalıştıkları yapay zekâ teknolojisi/ürünleri ve yapay zekâ silahlanma savaşları yerine, toplumların ortaklaşa geliştirdiği ve tüm insanlığın erişimine sunduğu yapay öğrenme teknolojilerinin geliştirilmesi için açık üretken yapay zekâ konusunda çalışmalar yapılmalıdır.

Önceki dönemlerde veri egemenliği konusu tartışılırken bugün onunla birlikte yapay zekâ egemenliği konusu da tartışılmaya başlanmıştır. Google, Microsoft, Meta, Apple gibi devasa büyük işletmelerin neredeyse tüm insanlığın verisine sahip olmasına karşı oluşan anarşik durum Blockchain teknolojisinin, Web3 ekosisteminin ortaya çıkmasına neden olmuştur. Sözü edilen küresel işletmelerin yapay zekâ egemenliğine karşı insanlığın nasıl bir toplumsal örgütlenme gerçekleştirebileceği yakın zamanda görülecektir. Aslında bunun ilk belirtileri açık kaynak kodlu dil modellerinde ve açık kaynak kodlu difüzyon modellerinde görülmeye başlanmıştır. Ancak giriş bölümünde ifade edilen temel (*foundation*) modellerin ne tür veri kümeleri

ile nasıl oluşturulduğu konusu asıl sorunu oluşturmaktadır.

Kurum ve kuruluşların üretken yapay zekâ ve çok modlu üretken yapay zekâ uygulamalarını ölçeklenebilir seviyede kullanabilmeleri için mevcut bilgi işlem altyapılarında yeni mimari çözümler kurgulamaları gerekmektedir. Geomatik/Harita mühendislerinin, mekânsal bilişim alanında çalışanların bağlamdan ayrılmadan daha yüksek doğrulukta sonuçlar üretebilen çok modlu üretken yapay zekâ uygulamalarını iş süreçlerine uyumlu hale getirecek planlamaları ve bu planlara uygun yatırımları yapmaları gerekmektedir.

Yapay zekâ çalışmalarındaki gelişmelere paralel olarak Geomatik/Harita mühendislerinin üretim bantları da diğer sektörlerde olduğu gibi değişecek ve yenilenecektir. Fotogrametri gibi ölçme temelli yaklaşımlardan üretken yapay öğrenme modeli temelli bilgisayarlı görü yaklaşımlarına doğru geçişler olacaktır. Yakın zamana kadar dar yapay öğrenme modelleri ile uydu ve fotogrametrik görüntülerinden nesne tespiti, sınıflandırma ve bölütleme gibi geleneksel 2B bilgisayarlı görü görevleri etkin olarak gerçekleştirilmekteydi. Işığın parlaklık değeri yerine ışığın yayılma özelliğini dikkate alan yeni yaklaşımlarla bir nesnenin yeni görünümlerini sentezleme ve görüntülerden 3B şeklini gerçekçi biçimde yeniden oluşturma görevleri gerçekleştirilmeye başlanmıştır. Böylece kullanıcının akıllı telefonundaki kameradan elde edilen görüntüler, Google StreetView gibi görüntü tabanlı uygulamalardaki görüntüler vb. görüntüler kullanılarak kentlerin yüksek kaliteli 3B temsilleri özel donanım, yazılım, uzmanlık, büyük zaman ve finansal yatırım gerektirmeden çok kısa zamanda oluşturulabilir. Bu tür dijital ikizler daha sonra metaverse ortamlarında farklı kullanıcı düzeylerinde farklı amaçlarla etkin olarak kullanılabilir.

Görüldüğü üzere gelişim çok hızlı yaşanmaktadır. Yakın zamana kadar uzamsal/mekânsal zekâ (GeoAI) kavramı tartışılırken, bugün uzamsal/mekânsal zekânın üretken uzamsal/mekânsal zekâya (GenGeoAI) evrilmesi konusu öne çıkmaktadır. Üretken uzamsal zekâ kavramının getirdiği paradigma değişimi mekânsal bilişim uygulama alanlarını genişletecek ve mekânsal bilişim endüstrisinin büyümesine katkı sağlayacaktır.

Çıkar Çatışması Beyanı

Yazar, bu çalışmada bilinen ilgili herhangi bir finansal veya finansal olmayan çıkar çatışması olmadığını beyan eder.

Kaynaklar

Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P., & Hedman, P. (2023). Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 19697-19705).

Beurer-Kellner, L., Fischer, M., & Vechev, M. (2023). Prompting is programming: A query language for large language models. *Proceedings of the ACM on Programming Languages*, *7*(PLDI), 1946-1969.

Bruce, J., Dennis, M., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., Aytar, Y., Bechtle, S., Behbahani, F., Chan, S:, Heess, N., Gonzalez, L., Osindero, S., Ozair, S., Reed, S., Zhang, J., Zolna, K., Clune, J., Freitas, N., Singh, S., & Rocktäschel, T. (2024). Genie: Generative Interactive Environments. *arXiv preprint arXiv:2402.15391*.

Chen, Z., Wang, F., & Liu, H. (2023). Text-to-3D using gaussian splatting. *arXiv preprint arXiv:2309.16585*.

Coulom, R. (2007). Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games* (pp. 72-83). Berlin, Heidelberg: Springer Berlin Heidelberg.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Güney, C. (2019). Mekansal zekanın getirdiği paradigma değişimi. *Jeodezi ve Jeoinformasyon dergisi*, *6*(2), 128-142.

Hamdi, A., Melas-Kyriazi, L., Qian, G., Mai, J., Liu, R., Vondrick, C., Ghanem, B., & Vedaldi, A. (2024). GES: Generalized Exponential Splatting for Efficient Radiance Field Rendering. *arXiv preprint arXiv:2402.10128*.

Hoffmann, J, Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., & Sifre, L. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, *42*(4), 1-14.

Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luiten, J., Kopanas, G., Leibe, B., & Ramanan, D. (2023). Dynamic 3D gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020). NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, *65*(1), 99-106.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., & Chen, M. (2021). GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., & Chen, M. (2022). Point-E: A system for generating 3D point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*.

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Rabby, A. K. M., & Zhang, C. (2024). BeyondPixels: A comprehensive review of the evolution of neural radiance fields. *arXiv preprint arXiv:2306.03000*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Song, H., Dong, L., Zhang, W. N., Liu, T., & Wei, F. (2022). CLIP models are few-shot learners: Empirical studies on VQA and visual entailment. *arXiv preprint arXiv:2203.07190*.

Subramanya, S. J., Devvrit, F., Simhadri, H. V., Krishnawamy, R., & Kadekodi, R. (2019). DiskANN Fast accurate billion-point nearest neighbor search on a single node. *Advances in Neural Information Processing Systems*, *32*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need (v7). *arXiv:1706.03762v7*.

Wu, S., Fei, H., Qu, L., Ji, W., & Chua, T. S. (2023). NExT-GPT: Any-to-any multimodal LLM. *arXiv preprint arXiv:2309.05519*.

Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., & Wang, X. (2023). 4D gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2020). XLNet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

Yi, T., Fang, J., Wang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., & Wang, X. (2023). GaussianDreamer: Fast generation from text to 3D gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*.

Zwicker, M., Pfister, H., Van Baar, J., & Gross, M. (2002). EWA splatting. *IEEE Transactions on Visualization and Computer Graphics*, *8*(3), 223-238.

URL-1: https://cvpr2022-tutorial-diffusion-models.github.io (Erişim Tarihi: 17 Ocak 2024).

URL-2: https://timbr.ai/blog/introducing-timbr-sql-kg (Erişim Tarihi: 30 Mart 2024).