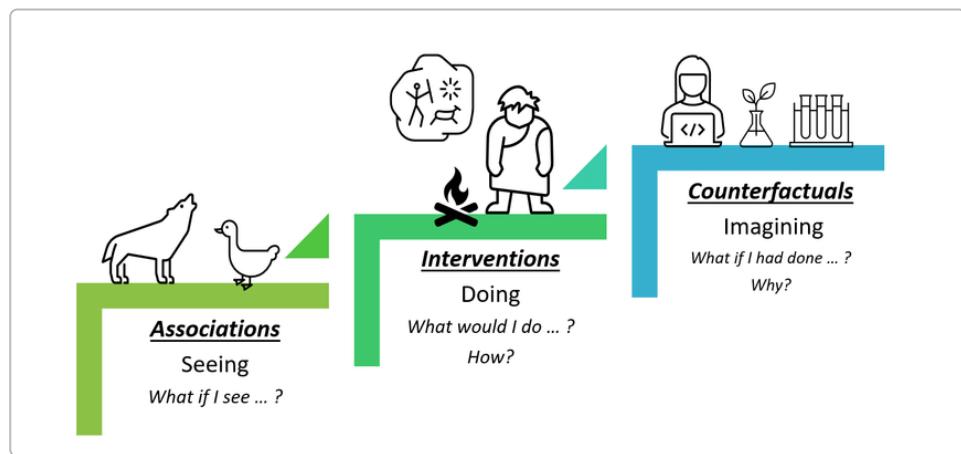


## Transformer Tabanlı LLM'lerde Nedensel Çıkarım



Pearl'ün “nedensellik merdiveni” (Ladder of Causation) gözlem, müdahale ve karşıolgusal olmak üzere üç düzeyi tanımlar. İlk basamak, salt gözlemsel veriden korelasyon çıkarma (association); ikinci basamak, sistem üzerinde müdahale (do) edip etkilerini görme; üçüncü basamak ise gerçekleşmemiş senaryoları düşünerek “ya şöyledi olsaydı?” sorularını yanıtlama (counterfactual) düzeyidir <sup>1</sup>.

### Arka Plan: Pearl, Rubin ve Yapısal Modeller Çerçeveesi

Nedensel çıkarım (causal inference) alanında Judea Pearl'ün **Yapısal Nedensel Modeli (SCM)** ve Donald Rubin'in **Potansiyel Sonuçlar Çerçeveesi** (Rubin Nedensel Modeli) temel yaklaşımlar olarak kabul edilir. Pearl'ün SCM yaklaşımı, nedensel ilişkileri **yönlendirilmiş döngüsüz grafikler** (DAG) ve “do-operatörü” ile ifade ederek müdahale ve karşıolgusal analiz imkanı sunar. Rubin'in çerçevesi ise **potansiyel sonuçlar/çıktılar** kavramını kullanarak tedavi (mihadale) ve kontrol durumlarındaki ortalama etkiyi (örn. ATE, CATE) istatistiksel olarak tanımlar. Her iki yaklaşım da nedensel kavramları, metrikleri ve analiz yöntemlerini sistematik biçimde tanımlamıştır <sup>2</sup>. Ancak bu geleneksel yöntemler çoğunlukla **yapısal veriler** (örn. tablo verileri) üzerinde çalışır ve doğal dildeki neden-sonuç ilişkilerini ele almakta yetersiz kalır <sup>3</sup>.

Öte yandan büyük dil modelleri (LLM) devasa miktarda metin verisi üzerinde eğitim alarak dildeki kalıpları ve ilişkileri öğrenir. Fakat bu eğitim verisi genellikle **nedensel işaretler içermeyen, yapılandırmamış metinler** olduğu için LLM'ler gördükleri ilişkilerin altında yatan gerçek nedenselliği anlamakta zorlanır <sup>4</sup>. Model, eğitimde karşılaştığı nedensel ifadeleri tekrarlayabilir fakat çoğunlukla bunları derin bir nedensel akıl yürütme ile değil, istatistiksel birlikte olarak “papagan gibi” yansıtır <sup>5</sup> <sup>6</sup>. Özellikle **karşıolgusal senaryolar** LLM'ler için güçlük arz eder: “eğer X olsaydı Y değişir miydi?” türü sorular, bazı değişkenleri sabit tutup diğerlerini değiştirmeyi gerektirdiğinden, salt ardışık kelime olasılıklarına dayanan Transformer modellerine doğal gelmez <sup>7</sup>. Sonuç olarak, mevcut büyük modeller **nedensel körlük** diyebeceğimiz bir durumdadır – metindeki korelasyonları yakalarken, **neden-sonuç ilişkilerini önceliklendirme** veya bu ilişkileri özel olarak öğrenme yeteneğine sahip degillerdir <sup>8</sup>. Bu eksiklik, karar destek veya politika analizi gibi **derin nedensel anlayış** gerektiren uygulamalarda LLM'lerin performansını sınırlamaktadır.

## LLM'lere Nedensel Akıl Yürütmeye Kazandırma Yöntemleri

Mevcut araştırmalar, Transformer tabanlı LLM'lere nedensel çıkarım yeteneği kazandırmak için çeşitli yöntemler önermektedir<sup>9</sup>. Aşağıda, öne çıkan yaklaşım ve modeller özetlenmiştir:

- **Prompt Mühendisliği ile Nedensel Düşünme:** En basit yöntem, **özel istemler (prompts)** tasarlayarak modele nedensel akıl yürütmemeyi tetiklemektir. Bu kapsamda doğrudan soru sormak, örnekler vererek **İçerik-odaklı öğrenme (ICL)** yapmak veya zincirleme düşünme telkini (**Chain-of-Thought, CoT**) vermek gibi stratejiler uygulanmıştır<sup>10</sup>. Örneğin, modele "Sen faydalı bir nedensel asistanın..." gibi bir rol tanımlaması yapmak, karmaşık nedensel sorularda performansı belirgin şekilde artırabilmektedir<sup>11</sup>. Causality özeline tasarılanmış istem şablonları da mevcuttur. **CausalCoT** adlı yaklaşım, adım-adım akıl yürütme stratejisini Pearl'ün nedensel çıkarım ilkeleriyle birleştirerek modeli önce sorudaki nedensel grafiği, nedensel soruyu ve ilgili verileri (ör. şartlı veya *do-olasılıkları*) tanımlamaya yönlendirir<sup>12</sup>. Bu sayede model, bir soruyu yanıtırken önce "hangi değişkenler ve ilişkiler söz konusu?" diye düşünüp ardından cevaba geçer. Nitekim, "**düşünce zinciri**" sağlayan bu tip istemler, modelin neden ve sonuçları açıkça ilişkilendirmesine yardımcı olur<sup>13</sup><sup>14</sup>. Ayrıca bazı araştırmacılar, LLM'lerde önyargıları gidermek amacıyla istemlere doğrudan nedensel analiz formülleri eklemiştir. Örneğin, **front-door** veya **back-door** düzeltme gibi Pearl'ün önerdiği istatistiksel tekniklerin istem düzeyinde uygulanmasıyla, model çıktılarındaki yanılıcı korelasyonları azaltma denemeleri yapılmıştır<sup>15</sup>. Sonuç olarak, prompt tabanlı yöntemler hızlı ve esnek şekilde modeli farklı nedensel görevlerde kullanmaya olanak tanır. Ancak bu yöntem, modeli özünde değiştirmediği için, tutarlılık sorunu veya her durumda güvenilir olmayan cevaplar üretilemesi gibi kısıtları da beraberinde getirir<sup>16</sup>.
- **Modeli İnce Ayar (Fine-Tuning) ile Eğitme:** LLM'lerin nedensel becerilerini geliştirmek için bir diğer yaklaşım, modeli özel nedensel görev verilerine göre **yeniden eğitmek (fine-tune)**dir. Örneğin Cai ve arkadaşları (2023), bir LLM'i **çiftler halinde nedensel keşif** (iki değişken arasındaki sebep-sonuç yönünü belirleme) görevine uyarlamak için sentetik bir veri kümesi oluşturmuştur<sup>17</sup>. Bu çalışma, doğrusal-gayıriGaussian yapıda sentetik veriler (Linear Non-Gaussian Acyclic Model, LiNGAM) kullanarak Mistral-7B modelini LoRA teknigiyle eğitti ve temel modele kıyasla anlamlı performans artışı elde etmiştir<sup>18</sup>. Benzer şekilde, **karşıolgusal veri oluşturma** yoluyla eğitimi zenginleştirmek de bir stratejidir: Modelin eğitim kümesine "kontrol değişkeni sabit tut,  $X$ 'i değiştir ve sonucu gözlemler" türünde sentetik örnekler eklemek, onun "what-if" mantığını kavramasını kolaylaştırabilir<sup>19</sup>. İnce ayar yaklaşımı, modele doğrudan nedensel bilgisayar öğretmeye yardımcı olsa da, büyük modellerin yeniden eğitiminin yüksek hesaplama maliyetli olduğu ve aşırı uyum (overfitting) riski taşıdığı bilinmelidir<sup>20</sup>. Dolayısıyla, bu yöntemde **veri setinin kalitesi** (çeşitli senaryoları kapsaması, gürültüsüz ve doğru nedensel etiketler içermesi) kritik önemdedir. Gerekirse, mevcut eğitim verisini **yeniden yapılandırmak veya etiketlemek** gündeme gelebilir – örneğin, kullanıcı sistemindeki CSV formatlı sohbet verilerine nedensel ilişkileri belirten etiketler eklemek veya ek eğitim için özel nedensel soru-cevap çiftleri derlemek gibi.
- **Geleneksel Nedensel Yöntemlerle Hibrit Modeller:** LLM'lerin dil anlama gücünü, klasik nedensel çıkarım algoritmalarının sayısal becerileriyle birleştirmek umut vadeden bir yaklaşımdır. Bazı çalışmalarda LLM, tek başına tüm nedensel muhakemeyi yapmak yerine, **yardımcı bir modül** gibi kullanılır. Örneğin Ban ve arkadaşları (2023), LLM'den çıkan metinsel çıkarımları **MINOBx** veya **CaMML** gibi grafik tabanlı nedensel keşif algoritmalarıyla birleştirerek hem salt LLM'den hem de salt veri odaklı algoritmdan daha başarılı sonuçlar elde etmiştir<sup>21</sup>. Bu tarz bir hibrit modelde LLM, insan dilindeki ipuçlarını yakalayıp bir nedensel hipotez önerirken; geleneksel algoritma, veriye dayalı sayısal bir doğrulama yapar. Benzer bir başka çalışma, LLM'yi adım adım bir **graf keşif aracı** olarak kullanmıştır: Jiralerspong ve arkadaşları (2024), bir LLM'in

her seferinde “hangi değişken diğerine etki eder?” sorusunu yanıtlayarak kademeli biçimde tüm nedensel grafiği oluşturmasını sağlamışlardır<sup>22</sup>. Geniş kapsamlı bir graf için olası kenar sayısını büyük ölçüde azaltan bu yöntem, dil modelinin içsel bilgisini kullanarak **aramayı sıralı biçimde daraltmıştır**. Üstelik gerçek gözlem verisi olmadan dahi çalışabilen bu yöntem, gerektiğinde istatistiksel verilerle de desteklenerek daha doğru sonuçlar verebilmiştir<sup>23</sup>. Hibrit yaklaşım, karar-destek sistemlerinde güvenilir nedensel ilişkiler bulmak için özellikle değerlidir; zira LLM bir uzman gibi hipotez kurarken, geleneksel yöntem bunu sınar veya quantifiye eder. Bununla birlikte, dil modeli ile istatistiksel modelin sonuçlarını **uyumlu şekilde birleştirmek** bazen zordur ve tutarsızlıkların giderilmesi araştırma konusu olmaya devam etmektedir<sup>24</sup>.

- **Nedensel Ajanlar ve Araç Kullanımı:** LLM’leri birer **araç kullanan yapay ajan** şeklinde konumlandırmak da mümkündür. Bu senaryoda model, bir soru karşısında kendi bilgi sınırlarını bilir ve uygun gördüğü takdirde harici bir **nedensel çıkarım aracını** (yazılım kütüphanesi, API, vb.) çağrıarak hesaplama yapır. 2024 yılında tanıtılan **Causal Agent** ve **Causal-Copilot** sistemleri, LLM’i merkezde tutup onun etrafına nedensel analiz yetenekleri eklemek üzere geliştirilmiştir<sup>25</sup>. Örneğin, böyle bir ajan model bir sağlık sorusunu yanıtırken otomatik olarak bir istatistiksel analiz modülünü çalıştırıp, ilgili veriseti üzerinde **arka-kapı düzeltme** uygulayabilir veya bir **nedensel etki büyülüğu** hesaplayabilir. Bu sayede geleneksel yöntemlerin çözebildiği daha karmaşık nedensel sorular da LLM destekli bir asistan tarafından cevaplanabilir hale gelir<sup>25</sup>. Araç kullanan yaklaşımın en önemli avantajı, LLM’nin zayıf olduğu sayısal hesaplama veya grafik çıkarma işini uzman bir sisteme devrederek **hata payını azaltması** ve ölçeklenebilirlik sağlamaasıdır. Nitekim Pawłowski ve ekibinin (2023) önerdiği yöntemde iki tür bilgi takviyesi test edilmiştir: (1) **Bağlam artırma** – soruya ilişkin önbilgileri genişletmek için LLM’ye istem içinde ilgili **nedensel grafik** ve bilinen **tedavi etkisi (ITE)** değerleri sunulmuştur; (2) **Araç kullanımı** – LLM, bir uzman sistemi API’na bağlanarak hesaplamaları ona yaptırmıştır. Bu iki yaklaşımından özellikle API çağrıma yöntemi, problem boyutu arttıkça daha tutarlı sonuçlar vermiştir<sup>26</sup> <sup>27</sup>. Benzer şekilde, Zhang ve arkadaşlarının LACR modeli, LLM’e geniş bir bilimsel metin korpusundan **olası nedensel bilgileri getiren bir arama mekanizması** entegre etmiştir<sup>28</sup>. Böylece model, kendi bilgi tabanında olmayabilecek nedensel bağlantıları, dış kaynaklı dokümanlardan çekerek kullanabilmiştir (Retrieval-Augmented Generation). Özette, LLM tabanlı bir ajan sistem, karar destek bağlamında gereken hesaplamaları ve uzmanlıkları uygun araçlara devrederek daha güvenilir nedensel akıl yürütme yapabilir.
- **Bilgiyle Zenginleştirilmiş Eğitim ve Temsiller:** Nedensel çıkarım kabiliyeti kazandırmanın bir diğer boyutu da, modelin ön eğitim veya ince ayar aşamasında **nedensel bilgiyle donatılması**dır. Bu, veriye nedensel etiketler eklemek veya modelin temsil uzayını nedenselliği yansıtacak şekilde değiştirmek anlamına gelir. Örneğin, son çalışmalar Transformer modellerinin gömülü uzaylarında **sebepsel ve nedensel olmayan faktörleri ayırt etmeyi** amaçlayan yöntemler geliştirmiştir<sup>29</sup> <sup>30</sup>. **Nedensel temsil öğrenimi (causal representation learning)**, modelin içsel olarak “X, Y’ye neden olur mu?” sorusuna duyarlı vektörler öğrenmesini hedefler. Ayrıca, **önyargıları gidermeye yönelik nedensel yaklaşım**lar da literatürde yer alır. *Causal-Debias* gibi çerçeveler, modelin eğitimi sırasında nedensel müdahaleler (interventions) ve **değişmez risk minimizasyonu (IRM)** uygulayarak, metindeki demografik yanlışlıkların veya spurious korelasyonların modele sirayet etmesini engellemeye çalışmıştır<sup>30</sup>. Bu tarz yöntemler, modelin salt korelasyonlara dayanarak karar vermesini engelleyip gerçek nedensel faktörlere odaklanması sağlanmaya yönelikdir. Hatta yakın zamanda **CausalINLP** veya **CausaLM** diyeboleceğimiz, nedensellik farkındalığına sahip temel modeller araştırılmaktadır<sup>31</sup>. Bu modeller, eğitimlerinin erken aşamalarından itibaren (token embedding ve pre-training safhası dahil) nedensel yapıları öğrenmeleri için tasarılanır. Örneğin, bazı çalışmalar grafik yapılı sentetik cümleler veya nedensel bilgi grafikleri ile birlikte ön eğitim yaparak LLM’lerin parametrelerine **nedensel evrenin kurallarını işleme fikrini** ortaya atmıştır<sup>32</sup> <sup>33</sup>. Bu, henüz gelişmekte olan bir

alan olsa da, gelecekte LLM'lerin yalnız dil bilgisini değil aynı zamanda **nedensel mantık yürütme kurallarını** da içselleştirecek şekilde eğitilebileceğine dair işaretler vardır<sup>31</sup>.

## Eğitim Verisinin Yeniden Yapılandırılması Gerekliliği

LLM'lere nedensel çıkarım yetisi kazandırmak, çoğunlukla eğitim verilerinde ve hedef görev tanımlarında bazı düzenlemeler gerektirebilir. Eğer mevcut model yalnızca **diyalog formatında** (system/user/assistant tarzı) genel amaçlı eğitildiyse, muhtemelen verideki nedensel ilişkiler açıkça etiketlenmemiş veya vurgulanmamıştır. Bu durumda, modeli nedensel düşünmeye yönlendirmek için **ek veri hazırlığı** yapmak gerekebilir. İki temel yaklaşım öne çıkmaktadır:

- **Nedensel Etiketleme ve Veri Zenginleştirme:** Mevcut veri kümenizi gözden geçirip, içinde geçen belirgin sebep-sonuç ilişkilerini işaretlemek bir seçenekdir. Örneğin, diyaloglarda kullanıcı ile asistan arasında geçen ve “*neden...?*” veya “... *olursa ne olur?*” biçimindeki etkileşimleri tespit edip bunları özel olarak etiketleyebilirsiniz. Ardından modeli, bu etiketli veriye dayalı olarak tekrar eğitmek, nedensel sorgulara duyarlığını artıracaktır. Eğer elinizde yeterli etiketli örnek yoksa, literatürdeki **nedensel soru-cevap veri setlerinden** yararlanmak değerlidir. Örneğin, **CausalQA (2022)** adlı kapsamlı veri seti, web'den derlenmiş **1,1 milyon** nedensel soru içerir ve hem nedensel keşif (CD) hem de nedensel açıklama (CE) sorularını kapsar<sup>34</sup>. Bu tür veri setlerini ince ayar için kullanmak, modelinizin benzer bağlamlardaki performansını iyileştirebilir. Benzer şekilde, **WikiWhy** veya **CausalBank** gibi kaynaklar da metin içerisindeki neden-sonuç ifadelerini barındıran örnekler sunar. **Karşıolgusal veri** oluşturma da veri zenginleştirmenin özel bir alt dalıdır: Örneğin, bir diyalogu alıp kilit bir faktörü tersine çevirerek (“X oldu” yerine “X olmasaydı”) yeni bir karşıolgusal senaryo üretebilir ve modelin bu değişikliğe tepkisini eğitebilirsiniz. Bu sayede model, alternatif dünyalarda ne olacağını düşünme alıştırması yapar<sup>7</sup>. Özette, eğer mevcut eğitim veriniz nedensel çıkarım açısından zayıfsa, onu **yeni örnekler, etiketler veya senaryolarla yeniden yapılandırmak** öğrenme kalitesini yükseltecektir.
- **Özel Amaçlı Veri Kümeleri ile İnce Ayar:** Akademik çevrede, LLM'lerin nedensel yeteneklerini ölçmek ve geliştirmek için son yıllarda çeşitli **benchmark** veri kümeleri oluşturulmuştur<sup>35</sup>. Modelinizi bu özel veri setleriyle eğitmek veya en azından test etmek, nedensel çıkarım konusundaki güçlü ve zayıf yönlerini görmeyi sağlar. Örnek olarak, **COPA (Choice of Plausible Alternatives)**, toplam 1000 adet neden-sonuç sorusu içeren ve modelin bir olayın olası nedenini ya da sonucunu ikili seçenek arasından seçmesini gerektiren bir benchmark'tır<sup>36</sup>. COPA, ortak akıl (commonsense) nedensellliğini ölçer ve bir LLM'in basit neden-sonuç bağlantılarını ne derece anladığını gösterir. **E-CARE (2022)** ise ~21 bin soruluk ve gündelik durumlara dair nedensel keşif ve açıklama görevlerini içeren bir veri setidir<sup>36</sup>. Daha uzmanlık gerektiren konularda, örneğin tıp ya da biyolojide, küçük ölçekli klasik testler de kullanılır: **“Asia” akciğer hastalığı ağı** (1988'den klasik bir 8 değişkenli nedensel model) veya **Sachs'in biyolojik ağı (2024)** gibi mini veri kümeleri, modeli çok net tanımlı nedensel grafik senaryolarında sınar<sup>36</sup>. 2024 yılında yayınlanan **CausalBench**, farklı kaynaklardan derlenen 15 ayrı veri kümесini bir araya getirerek korelasyon ve nedensellik ayrimını ölçmeyi hedefleyen bir benchmark sunmuştur<sup>34</sup>. Yine 2024'te Chen ve arkadaşlarının hazırladığı **CaLM-Bench**, bilim ve genel kültür alanlarında ~280 bin soruya modelin nedensel atif (CA) yapabilme ve keşif (CD) kabiliyetini değerlendirir<sup>34</sup>. Bu benchmark'lar hem birer ölçüm aracı hem de gerekirse ince ayar verisi olarak kullanılabilir. Dolayısıyla, modelinizi karar destek veya QA ortamında kullanmadan önce, bu tür nedensel odaklı veri setlerinde sınamak ve gerektiğinde bu veriyle **yeniden eğitmek** faydalı olacaktır.

## Karar Destek ve Soru-Cevap (QA) Bağlamında Nedensel Çıkarım Stratejileri

Transformer tabanlı modellerin nedensel çıkarım becerileri, özellikle **karar destek sistemleri** ve **soru-cevap** uygulamalarında değer kazanmaktadır. Bu bağamlarda başarılı olmak için önerilen stratejiler söyle özetlenebilir:

**1. Karar Destek için Nedensel Modelleme:** Karar destek sistemleri (örneğin tıbbi teşhis, finansal risk analizi veya politika simülasyonu) genellikle *"bir faktör değişirse sonuç ne olur?"* sorusuna yanıt arar. LLM'ler bu tür durumlarda, kullanıcıya tavsiye vermeden önce bir *nedensel akıl yürütme zinciri* takip etmelidir. Uygulanabilir bir strateji, modelin karar öncesi **kendi kendine açıklama** yapmasını sağlamaktır. Örneğin, bir tıbbi karar destek asistanı, *"Bu ilaçın verince hastada kan basıncı düşer çünkü ilaçın beta-blocker etkisi vardır"* gibi nedensel bir açıklamayı zincirleme düşünceyle üretip, sonra nihai öneriyi sunabilir. Bu, modele **nedensel açıklama görevi** verilerek teşvik edilebilir. Nitekim araştırmalar, modelin bir olayı açıklamasını istemenin, onun nedensel bağlantıları daha doğru kurmasına yardımcı olabileceğini göstermektedir<sup>37</sup> <sup>38</sup>. Karar destek bağlamında bir diğer kritik nokta, **counterfactual (karşıolgusal) analiz** yapabilmektir. Örneğin, asistanınıza *"Hasta 10 yıl daha genç olsaydı tedavi değişir miydi?"* diye sorulduğunda, model bu hipotetik durumu değerlendirebilmelidir. Bu amaçla, modele belirli formatlar öğretebilirsiniz: Karşıolgusal sorular geldiğinde önce mevcut durumu özetle, sonra değişkeni değiştir ve sonucu kıyasla gibi bir yaklaşım izletebilirsiniz. Hatta imkan varsa, modelinizin arka planda bir **simülasyon aracı** çalıştırmasını sağlayabilirsiniz (yukarıda bahsedilen araç kullanan ajan yapısı ile). Örneğin ekonomi alanında, LLM bir politika değişikliğinin etkisini anlamak için bir nedensel keşif kütüphanesini tetikleyip, veri üzerindeki beklenen etkiye hesaplararak cevaba dahil edebilir. Bu tür **entegre yaklaşımlar**, karar destek sisteminin tavsiyelerini daha güvenilir ve gerekçeli hale getirir. Son olarak, karar destekte model yanıtlarının tutarlılığı çok önemlidir; bu nedenle model, benzer nedensel senaryolarda çelişen cevaplar vermemesi için **kural tabanlı ince ayar** ile denetlenmelidir (örneğin, aynı nedensel ilişkisi farklı cümlelerle sorunca hep aynı yönde cevap vermesi beklenir).

**2. Soru-Cevap Sistemlerinde Nedensel Sorgular:** Klasik QA sistemleri genellikle olgusal (faktöyel) soruları yanıtlar; ancak kullanıcılar **"neden?"** ve **"nasıl olur?"** türü sorularla da sıkılıkla ilgilenir. LLM tabanlı bir QA sisteminin nedensel sorulara yanıt verebilmesi için, model hem dil becerilerini hem de arka plandaki nedensel bilgiyi kullanmalıdır. **Ortak akıl (commonsense) nedenselliği** için COPA gibi testlerde başarılı performans göstermek bir başlangıç metrik olabilir. Örneğin, *"Buzlu yolda hızlı araba kullanmak neden tehliklidir?"* sorusuna, *"çünkü fren mesafesi artar ve kaza riski yükselir"* şeklinde, arkaplandaki fiziki nedenselliği açıklayan bir cevap ideal olmalıdır. Modelinizi bu tip sorular için eğitirken, **zincirleme düşünce istemlerini** QA ile birleştirilebilirsiniz. Yani, modelden doğrudan yanıt istemek yerine *"Adım adım düşünerek cevapla."* tarzı yönlendirmeler, onun önce olası nedenleri tartıp sonra en uygununu seçmesine yardımcı olur<sup>14</sup>. Nitekim sıfır atış zincirleme düşünce (0-shot CoT) yönteminin, özellikle çıkışlımsal QA sorularında başarıyı artırdığı rapor edilmiştir<sup>39</sup> <sup>40</sup>. Soru-cevap sistemlerinde bir diğer strateji, **belge veya bilgi getirme (RAG)** ile nedensel cevabı desteklemektir. Kullanıcı sorusu spesifik bir alandaysa (örneğin "Küresel ısınmanın X bölgesindeki kuraklığa etkisi nedir?"), model önce ilgili bilimsel veya haber metinlerini arayıp, oradaki nedensel açıklamaları çekebilir ve bunları kendi cümleleriyle harmanlayarak yanıt verebilir. Bu şekilde, LLM hem güncel ve doğrulanmış bilgi kullanır hem de nedensel bağlam hatalarını azaltır. Son olarak, QA sisteminde cevapların **gerekçelendirilmesi** önem taşır. Model her **neden** sorusuna yanıt verirken mümkünse **kanıt veya kaynak belirtecek şekilde** (sizin durumunuzda, belki model cevabına eğitim verisindeki bir referansı ekleyebilir) cevap üretmelidir. Bu, Rubin'in "karşılaştırma dünyaları" yaklaşımına da uygundur: Bir şeyi neden dediğini, alternatifini düşündüğünde sonuç nasıl olurdu diyerek açıklamak, cevabı daha anlaşılır kılar. Örneğin, *"Çünkü A olmasaydı B de olmazdı; zira veriler A'nın B üzerindeki etkisini gösteriyor"* gibi bir cevap, kullanıcıya nedensel sonuç ilişkisini net biçimde aktaracaktır. Bu tür yanıt formatlarını modelinize öğretmek için, ince ayar

esnasında **açıklamalı cevapları** (explanations) içeren veri kullanabilirsiniz. **CausalExplanation (CE)** tarzi etiketler taşıyan veri setleri (örneğin **WikiWhy** veya **E-CARE** içindeki açıklama kısımları) bu iş için uygundur.

## Örnek Uygulamalar, Araştırmalar ve Araçlar

Nedensel çıkarım odaklı LLM geliştirme, akademide ve endüstride hızla büyüyen bir araştırma alanıdır. Bu bağlamda öne çıkan güncel çalışmalar ve kaynaklardan bazıları şunlardır:

- **CausalLM ve Araştırma Makaleleri:** 2025 itibariyle LLM'ler ve nedensellik konusunda kapsamlı bir literatür oluşmuştur. Jing Ma (2025) tarafından derlenen “*Causal Inference with Large Language Models: A Survey*”<sup>41</sup> bu alandaki ana eğilimleri ve görevleri özetleyen güncel bir çalışmıştır. Benzer şekilde Zhijing Jin ve diğerlerinin çalışmaları, LLM'lerin nedensel bilgisini ölçmek ve geliştirmek üzerine yoğunlaşmıştır. Örneğin Jin ve arkadaşları (2023), **CausalCoT** stratejisile LLM'lerin Pearl'ün neden-sonuç hesaplamalarına benzer şekilde yanıtlar verebildiğini gösteren deneyler yapmıştır<sup>42</sup>. Başka bir araştırma, **Zero-shot Causal Graph Extrapolation** yöntemiyle, yalnızca değişken isimlerini vererek LLM'nin olası nedensel ağ yapısını çıkarabildiğini ileri sürmüştür<sup>43</sup>. Bu tür çalışmalar, LLM'lerin içsel olarak ne kadar nedensel bilgi barındırdığını ve uygun yönlendirmeye neler başarabileceğini keşfetmektedir.
- **Github Projeleri ve Araçlar:** Uygulamaya yönelik olarak, topluluk birkaç açık kaynak proje geliştirmektedir. **CausalAgent** (Han vd., 2024) adlı Github projesi, bir LLM etrafında inşa edilen ve **DoWhy** gibi popüler nedensel çıkarım kütüphanelerini araç olarak kullanan bir ajan mimarisini sunar. Bu ajan, verilen bir veri kümesinde otomatik olarak nedensel keşif yapıp LLM'nin diliyle sonuçları raporlayabilir. Benzer bir proje olan **Causal-Copilot** (Wang vd., 2024), araştırmacılar ve veri bilimciler için etkileşimli bir nedensel analiz asistanı sağlamayı hedefler<sup>25</sup>. Bu araç, kullanıcıdan doğal dilde aldığı nedensel soruları uygun istatistiksel yöntemlerle eşleştirerek cevaplamakta, gerektiğinde sonuçları grafik veya tablo şeklinde sunmaktadır. Ayrıca **Awesome-Causal-LLM** gibi Github repo'ları, bu alandaki makaleleri ve model kontrol listelerini derleyerek araştırmacılara yol göstermektedir<sup>44</sup>. Eğer kendi LLM modelinizi geliştirmeye devam edecekseniz, bu tür kaynaklardaki örnek istem şablonları, değerlendirme metrikleri ve görev tanımlarından yararlanabilirsiniz.
- **Benchmark Veri Setleri ve Değerlendirme:** Yukarıda bahsedilen veri setlerine ek olarak, **CaLM (Causal Language Modeling) Benchmark** gibi kapsamlı testler de mevcuttur. CaLM (Chen vd., 2024) veri kümesi, Pearl'ün nedensellik merdivenindeki tüm basamakları (associational, interventional, counterfactual) kapsayan çeşitli sorulardan oluşur ve yaklaşık 126 bin örnek içerir<sup>34</sup>. Bu benchmark, LLM'lerin sadece korelasyon mu yoksa gerçek nedensellik mi yakaladığını ölçmek için tasarlanmıştır. Örneğin model, CaLM içinde “X olduğu için mi Y oldu?” tarzı bir soruya doğru yanıt verebilmelidir; ardından “X olmasaydı Y olur muydu?” şeklindeki karşılıklı takibi de tutarlı cevaplamalıdır. **CausalBench (2024)** ise sağlık başta olmak üzere farklı alanlardan derlenmiş veri setleriyle LLM'leri çok yönlü test eder<sup>34</sup>. Bu benchmark'lar, sadece model geliştirme aşamasında değil, aynı zamanda farklı modelleri nedensel performans açısından karşılaştırmak için de kullanılmaktadır. Nitekim literatürde GPT-4 gibi gelişmiş modellerin bu testlerde yüksek başarı gösterdiği, ancak bazen halen yüzeysel dil kalıplarına dayalı hatalar yapabildiği rapor edilmiştir<sup>45</sup>. Kendi modelinizi bu benchmark'larla kıyaslamak, hangi tip nedensel sorularda zorlandığını ortaya koyacak ve iyileştirme için yol haritası sunacaktır.
- **Özel Amaçlı Framework ve Kütüphaneler:** Doğrudan LLM eğitimi veya etkileşimi için tasarlanmış nedensel kütüphaneler de gelişmektedir. Örneğin, Microsoft'un **DoWhy** ve **EconML**

kütüphaneleri, Python ortamında nedensel etki analizi yapmayı kolaylaştırır ve LLM tabanlı ajanlara entegre edilebilir. **CausalNLP** adı altında bazı araştırmacılar, metin verisinden nedensel çıkarım yapmaya yardımcı olacak araçlar sunmuştur (ör. metindeki potansiyel nedensel ifadeleri çıkarıran modüller). **PyTorch** ekosisteminde de “causal transformer” denen deneysel eklentiler görülmektedir – örneğin, bir Transformer’ın attention mekanizmasını *do-calculus* kurallarına göre kısıtlamaya çalışan veya belirli nöronları “nedensel değişken” şeklinde uzmanlaştıran araştırmalar vardır. Bu tarz framework’ler henüz olgunlaşma aşamasında olsa da, ileride **tamamen nedensellik odaklı dil modellerinin** ortaya çıkması beklenmektedir<sup>31</sup>. Son olarak, **Counterfactual Data Augmentation** araçlarından bahsetmek yerinde olur: Örneğin *DiCE* veya *CounterfactualGAN* gibi araçlar, eldeki veriyi küçük değişikliklerle yeniden yazarak karşıolgusal örnekler üretir. Bunlar, LLM’lerin eğitimine dahil edilirse, modelin daha sağlam ve nedensel değişime duyarlı olmasını sağlayabilir.

**Sonuç:** Transformer tabanlı LLM’lere nedensel çıkarım yeteneği kazandırmak, birden çok disiplinin kesiştiği zorlu fakat heyecan verici bir hedeftir. Pearl’ün grafiksel model yaklaşımı, Rubin’ın potansiyel sonuçlar modeli ve diğer yapısal yöntemler, doğru uygulandığında LLM’lerin **“neden?” sorusunu daha insan benzeri şekilde yanıtlamasını** mümkün kılacaktır. Bunu başarmak için hem veri düzeyinde (doğru senaryoları öğretmek) hem de model düzeyinde (doğru yapı ve araçlarla donatmak) çalışmalar yapılmalıdır. Güncel araştırmalar, istem mühendisliğinden model eğitimine, araç entegrasyonundan bilgi artırmaya kadar çeşitli yollarla bu entegrasyonu sağlamaya çalışıyor. Önümüzdeki dönemde, karar destek sistemleri ve QA uygulamalarında, yalnızca dilsel tutarlılığı yüksek değil aynı zamanda **nedensel tutarlılığı** da güçlü LLM’ler görmeyi bekleyebiliriz<sup>46</sup> <sup>47</sup>.

**Kaynaklar:** Cevap boyunca belirtilen kaynaklar makale ve araştırma sonuçlarından derlenmiştir. Özellikle NAACL 2025 Nedensel LLM Anketi<sup>41</sup> <sup>9</sup> ve ilgili çalışmalar, belirtilen iddiaların dayanağını oluşturmaktadır. Bu kapsamlı araştırmalar, LLM’lerin nedensel çıkarım basamaklarında nasıl performans gösterdiğini, hangi yöntemlerle iyileştirilebildiğini ve gelecekte hangi yönlere odaklanılması gerektiğini ortaya koymaktadır. Bu yanıtın hazırlanmasında adı geçen tüm kaynaklar ilgili satır referanslarıyla birlikte sunulmuştur. Bu sayede, her bir iddianın dayandığı orijinal çalışmaya erişerek derinlemesine inceleme yapmanız mümkün olacaktır.

---

<sup>1</sup> Pearl's Ladder of Causation. The first rung, associations, only allows... | Download Scientific Diagram  
[https://www.researchgate.net/figure/Pearls-Ladder-of-Causation-The-first-rung-associations-only-allows-predictions-based\\_fig2\\_357875366](https://www.researchgate.net/figure/Pearls-Ladder-of-Causation-The-first-rung-associations-only-allows-predictions-based_fig2_357875366)

[2](#) [3](#) [9](#) [10](#) [11](#) [16](#) [17](#) [18](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [45](#)

aclanthology.org

<https://aclanthology.org/2025.findings-naacl.327.pdf>

[4](#) [5](#) [6](#) [7](#) [8](#) [12](#) [13](#) [14](#) [15](#) [19](#) [29](#) [30](#) [31](#) [32](#) [33](#) [42](#) [43](#) [46](#) [47](#) Causality for Large Language Models

<https://arxiv.org/html/2410.15319v1>

<sup>44</sup> anpwu/Awesome-Causal-LLM - GitHub

<https://github.com/anpwu/Awesome-Causal-LLM>