

ÇEVİRİM İÇİ DİLBİLİM ARAŞTIRMALARI İÇİN PCIBEX FARM: TASARIM ve UYGULAMA

Oktay ÇINAR

Dr. Öğr. Üyesi, İstanbul Medeniyet Üniversitesi, Dilbilimi Bölümü, oktay.cinar@medeniyet.edu.tr, ORCID:
0000-0002-9822-7574

Çınar, O. (2025). Çevrim içi dilbilim araştırmaları için PCIBex Farm: Tasarım ve uygulama. İçinde O. Çınar, F. Başbuğ & H. Aydemir (Ed.), *Çağdaş dilbilim yazıları I: İMÜ Dilbilimi 15. yıl armağanı* (ss. 81-108). Artsürüm. <https://doi.org/10.7816/imuling-15-2025-01X004>

ÖZ

Son yıllarda deneysel dilbilim ve psikodilbilim alanlarında çevrim içi veri toplama yöntemlerine olan ilgi belirgin biçimde artmıştır. Bu eğilim, katılımcı erişimini kolaylaştıran ve deneysel süreçleri daha esnek hale getiren web-tabanlı platformların yaygınlaşmasıyla yakından ilişkilidir. Bu çalışma, PCIBex platformu ve PennController eklentisinin deneysel dilbilim araştırmalarında sunduğu olanakları kapsamlı biçimde tanıtmayı amaçlamaktadır. Öncelikle, PCIBex kullanılarak yürütülebilen başlıca deney türleri alanda yapılan çalışmalarla örneklendirilmektedir. Bu genel çerçevenin ardından, platformun somut kullanımını göstermek amacıyla örnek bir öz-denetimli okuma deneyi ayrıntılı biçimde sunulmaktadır. Uygulama kapsamında, Türkçede içe yerleşik tümcelerde boş ve açık özne adlarının öncüllerine gönderiminin işlenmesinin çevrim içi ortamda nasıl test edilebileceği gösterilmiştir. Deney tasarımı süreci; bilgilendirme ve gönüllü katılım onamı, yönergeler, katılımcı bilgi formu, alıştırma aşaması, ana test ve veri kaydı adımlarıyla sistematik biçimde ele alınmıştır. Ayrıca PennController tarafından üretilen olay-temelli sonuç dosyasının yapısı açıklanmış ve verilerin R ortamına aktararak betimleyici istatistikler aracılığıyla nasıl çözümlenebileceği gösterilmiştir. Çalışma, PCIBex'in çevrim içi deneysel dilbilim araştırmaları için erişilebilir, yeniden üretilebilir ve ölçeklenebilir bir altyapı sunduğunu ortaya koymaktadır.

Anahtar Sözcükler: PCIBex, PennController, Deneysel dilbilim, Öz-denetimli okuma, Çevrim içi deneyler

ABSTRACT

In recent years, experimental linguistics and psycholinguistics have increasingly adopted online data collection methods. This trend is closely related to the widespread use of web-based platforms that facilitate participant recruitment and enable flexible experimental designs. This study aims to provide a comprehensive overview of the PCIBex platform and the PennController extension in experimental linguistic research. First, the paper outlines and exemplifies the main types of studies that can be conducted

using PCIBex. Building on this overview, the study then presents a detailed example application to illustrate the practical use of the platform. Specifically, a self-paced reading experiment is introduced to demonstrate how the processing of the coindexation of null and overt subject pronouns in embedded clauses with their antecedents in Turkish can be investigated in an online environment. The experimental workflow is described step by step, covering informed consent, instructions, participant information forms, practice trials, the main test phase, and data storage. Special attention is given to the event-based structure of the results file generated by PennController and to the procedures for transferring and summarizing the data in the R environment using descriptive statistics. Overall, the study shows that PCIBex and PennController offer an accessible, reproducible, and scalable framework for conducting a wide range of online experimental studies in linguistics.

Keywords: PCIBex, PennController, Experimental linguistics, Self-paced reading, Online experimentation

1. Giriř

Son yıllarda dilbilim arařtırmalarında deneysel yöntemlerin kullanımını önemli bir artış göstermektedir. Bu yöntemler, dilin nasıl işlemlendiğini, üretildiğini ve algılandığını anlamak amacıyla tasarlanan kontrollü çalışmalar aracılığıyla, dilsel süreçlerin bilişsel yönlerini inceleme olanağı sunmaktadır. Öz-denetimli okuma, kabul edilebilirlik yargı testleri, sözcüksel karar verme ve işitme temelli testler bu deneysel yaklaşımlara örnek olarak gösterilebilir. Söz konusu bu çalışmalar genellikle bilgisayar ortamında yürütülmekte, katılımcıların tepkileri özel yazılımlar aracılığıyla ölçülmekte ve analiz edilmektedir. Bu yaklaşım, arařtırmacılara yüksek düzeyde kontrol imkanı sağlayarak güvenilir ve tekrarlanabilir sonuçlar üretme açısından önemli avantajlar sunmaktadır. Ancak bu avantajlara karşın, söz konusu çalışmalar katılımcı erişimi ve zaman yönetimi bakımından belirli sınırlılıklara sahiptir. Veri toplamak için kullanılan dilbilimsel deney araçları, genellikle aynı anda yalnızca tek bir katılımcı tarafından kullanılabilirliğinden, veri toplama süreci ardışık biçimde yürütülmektedir. Bu durum, çok sayıda katılımcıdan veri toplanması gerektiğinde sürecin oldukça zaman almasına yol açmaktadır. Ayrıca, bu çalışmaların belirli fiziksel koşullarda yürütülmesi gerekliliği, arařtırmacıların daha geniş ve çeşitlendirilmiş katılımcı gruplarına erişimini de sınırlandırmaktadır.

Bunun dışında, COVID-19 pandemisi ile birlikte bu sınırlılıklar daha da belirgin hale gelmiş ve yüz yüze veri toplama süreçleri büyük ölçüde güçleşmiştir. Bu gelişme, arařtırmacıları çevrim içi ortamda deney yürütmeye yönlendirmiştir. Bu gibi ihtiyaçlara yanıt olarak, web-tabanlı deney platformları, son yıllarda daha fazla kullanılmakta ve daha geniş bir katılımcı yelpazesine veri toplama olanağı sağlamaktadır. Bu platformlar arasında öne çıkan PCIBex (<https://farm.pcibex.net/>) tarayıcı üzerinden erişilebilen, çeşitli dilbilimsel çalışmaların yürütülebileceği, açık kaynaklı ve ücretsiz web-tabanlı bir deney platformudur. Bu açıdan, PCIBex, çeşitli

dilbilimsel deneylerin web-tabanlı yürütülebilmesi açısından arařtırmacılar için önemli kolaylıklar sunmaktadır.

Bu çalışmada, PCIBex platformunun temel altyapısı, dilbilimsel deney tasarımı süreçleri ve dilbilim alanındaki uygulamaları incelenecektir. Ayrıca, platformun hangi tür dilbilimsel çalışmalar için kullanılabileceği, arařtırmacılara sunduğu yenilikçi olanaklar tartışılacaktır. Bunun dışında çevrim içi deney yürütmenin getirdiği sınırlılıklara da değinilecektir.

Çalışmanın genel yapısı řu şekildedir. 2. bölümde, PCIBex platformunun tarihçesi, genel özellikleri ve getirdiği yenilikler ele alınacaktır. 3. bölümde, platform üzerinden yürütülebilecek dilbilimsel deney türleri, önceki çalışmalardan örnekler ve uygulama alanları tanıtılacaktır. 4. bölümde, PCIBex kullanılarak geliştirilen örnek bir deney tasarımı ayrıntılı biçimde sunulmakta; deneyin kurulum aşamaları, katılımcı yönetimi ve uygulama ve veri çözümleme süreci adım adım açıklanmaktadır. 5. ve son bölümde ise PCIBex'in deneysel dilbilim arařtırmalarındaki güçlü ve zayıf yönleri, sunduğu avantajlar ve çevrim içi deneylere özgü sınırlılıklar bütüncül bir değerlendirme çerçevesinde ele alınacaktır.

2. PCIBex: PennController, Arayüz ve Kullanım Özellikleri

2.1 PennController

PCIBex (internet tabanlı deneyler için PennController)^{1 2}, geniş ölçekli davranışsal deneylerin çevrimiçi olarak yürütülmesi amacıyla tasarlanmış, ücretsiz bir web tabanlı platformdur. Ibox, ilk olarak 2007 yılında Alex Drummond tarafından geliştirilmiş; 2018 yılında ise Jeremy Zehr ve Florian Schwarz tarafından PennController adlı bir kütüphane üzerine inşa edilerek genişletilmiş ve böylece Ibox'ın bir uzantısı hâline gelmiştir (Zehr & Schwarz, 2018).

Platform, 2010'dan 30 Eylül 2021'e kadar <https://spellout.net/ibexfarm> adresinde çalışmıştır. 31 Aralık 2020'den 30 Eylül 2021'e kadar ise yeni bir Ibox Farm sürümü <https://ibex.spellout.net> adresi üzerinden hizmet vermiştir. Güncel ve aktif web sitesi ise řu linktedir: <https://farm.pcibex.net/>

PennController, bir JavaScript tabanlı eklenti kütüphanesidir ve açık kaynak kodludur. Bu eklenti, Ibox'ın sınırlı deney yapısını (özellikle metin tabanlı görevleri) genişleterek, görsel, işitsel ve etkileşimli dilsel deney

¹“PennController” sözcüğünde “Penn” Pennsylvania Üniversitesi'nin kısaltması olarak kullanılmaktadır. İlk olarak Pennsylvania Üniversitesi tarafından desteklendiği için bu adla kısaltılmıştır. Dolayısıyla “Penn Denetleyici” olarak da Türkçeleştirilebilir; ancak çalışmada özel isim olarak kabul edilip orijinal adı ile kullanılacaktır.

² “PCIBex” kavramı ise “PennController for Internet-Based Experiments” (internet tabanlı deneyler için PennController) sözcüğündeki sözcüklerin ilk harfleri kısaltılarak oluşturulmuş bir sözcüktür.

tasarımlarının kolayca tasarlanmasını sağlar. Bu sayede, arařtırmanın basit sözdizimiyle karmařık deneyleri oluřturmasına olanak sağlar.³

PCIBex arayüzünde, PennController komutlarıyla deney nesnelere (ör. text, image, audio, button, keypress) tanımlanmaktadır. Ařağıda PennController kütüphanesi kullanılarak oluřturulmuř basit bir deney yapısı gösterilmektedir:

```
PennController.ResetPrefix(null);

newTrial("talimat",
  newText("Bu bir okuma testidir. Bořluk tuřuyla ilerleyin.")
  .print()
  ,
  newKey(" ")
  .wait()
);

Template("deneyListesi.csv", row =>
  newTrial("okuma",
    newController("DashedSentence", { s: row.cumle })
    .print()
    .log() // her kelimenin gösterilme süresi kaydedilir
    .wait()
  )
);
```

řekil 1. Örnek bir deney yapısı

PennController.ResetPrefix(null) komutu, kodun daha sade biçimde yazılmasına olanak tanır. newTrial() her bir deney aşamasını (örneğin talimat, alıřtırma veya uyarın gösterimi) tanımlar. newText() katılımcıya yazılı bir bilgilendirme sunarken, newKey() nesnesi katılımcının belirli bir tuřa (bu örnekte boşluk tuřu) basmasını tanımlar. Asıl deney kısmında ise benzer bir yapıyla newController("DashedSentence", {...}) nesnesi kullanılarak, bu örnekte cümleler öz-denetimli okuma biçiminde, yani sözcük sözcük sunulur. Her sözcüğe geçiřteki süre, katılımcının okuma tepki süresi (reaction time) olarak kaydedilir. .log() komutu, katılımcının yaptığı iřlemlere ait verileri kaydetmek için kullanılır. Her deneme sonunda bu veriler PCIBex'in sonuç dosyasına (.csv) otomatik olarak yazılır.

Yukarıda da görüldüğü gibi, standard Javascript dilinin kullanan PCIBex dili Penncontroller komutlarıyla basittir. Bunlar dıřında daha ileri seviye özellikle, iřitme testleri ve göz izleme (Papoutsaki et al., 2016) gibi uygulama programlama özelliklerine de sahiptir.

2.2 Arayüz ve kullanım özellikleri

Arařtırmacılar, platformun "Script Editor" (Kod Düzenleyici) bölümünde hazır řablonları kullanabilir veya kendi deney kodlarını sıfırdan oluřturabilirler. Arayüzün ekran görüntüsü řekil 2'de sunulmaktadır. Yapılan tüm deęiřiklikler sistem tarafından otomatik olarak kaydedilmektedir. Deneyin ön izlemesi, "Refresh" (Yenile) ile aynı sekmede veya "Open in new

³ <https://doc.pcibex.net/> baęlantısında deneylerin nasıl yürütüleceęi, komutlarının nasıl kullanıldıęı gibi temel kavramlar ve temel tutorial'a ulařılabilir.

tab” (Yeni sekmede aç) seçeneđiyle yeni bir sekmede yapılabilir. “Resources” (Kaynaklar) sekmesinde arařtırmacı, deneye ait dosyaları yükleyebilir (örn., uyarıların yer aldığı .csv tabloları, ek HTML sayfaları veya ses ve görsel dosyaları). “Scripts” (Kodlar) sekmesi yeni JavaScript dosyaları eklemeye, “Aesthetics” (Görsel Düzen) sekmesi ise deneyin tasarımını biçimlendirmeye olanak tanır. PennController kütüphanesi ise, “Modules” (Modüller) sekmesinde yer alır ve deneyin yürütülmesini sađlayan temel bileşendir.



Şekil 2. PCIBex arayüzü

Kod düzenleyicinin sađ kısmında yer alan “Actions” (Eylemler) paneli deney yönetimi için çeşitli işlevler içerir. “Results” (Sonuçlar) sekmesi, deneyin çıktı dosyasını ön izleme veya indirme olanađı sađlar. “Share” (Paylaş) sekmesi, deney için otomatik olarak bir paylaşım bağlantısı (link) oluşturur. Arařtırmacılar, PennController sözdizimiyle oluşturdukları deneyleri platformun arayüzü üzerinden kaydeder ve doğrudan web tabanlı bir deney bağlantısı olarak yayımlayabilirler. Bu bağlantı, deneyin test edilmesi, hata ayıklanması veya yayımlanan bir çalışmanın tekrarlanabilirliğini sađlamak amacıyla başkalarıyla paylaşılabilir. Veri toplama sürecine geçildiğinde, katılımcılara veri toplama bağlantısı gönderilir ve bu bağlantı üzerinden gelen sonuçlar sistemde ayrı olarak kaydedilir. Katılımcılar deney bağlantısını açtıklarında, herhangi bir ek yazılım ya da eklenti yüklemeye gerek kalmadan, tarayıcı üzerinden doğrudan deneye katılabilirler.

“Download” (İndir) seçeneđi, deney projesinin tamamını yerel bilgisayara indirme olanađı sunar. PCIBex, her katılımcının yanıtlarını otomatik olarak kaydeder. Tüm veriler sistemde oturum bazında depolanır ve arařtırmacı tarafından daha sonra .csv biçiminde indirilebilir. Elde edilen bu ham veriler, istatistiksel analiz yazılımlarında veya programlama dillerinde (ör. R, Python) çözümlenebilir.

“Git Sync” seçeneđi, mevcut bir Git deposunu projeye entegre etmeyi sađlar. “Settings” (Ayarlar) sekmesi ise deneye açıklama ekleme ve projenin kopyalanabilir olup olmadığını belirleme seçeneklerini içerir.

Sonuç olarak, PCIBex Farm arařtırmacıya deneyin tüm aşamalarını tek bir çevrim içi arayüz üzerinden yönetme olanađı sunar. Bu yönüyle platform, çevrim içi dilbilimsel ve psikodilbilimsel deneylerde erişilebilir, yeniden üretilebilir ve paylaşımına açık bir arařtırma ortamı sađlar.

3. Deneysel Dilbilim Bađlamında PCIBex Kullanımı

Bu bölümde, platform üzerinden yürütülebilecek dilbilimsel deney türleri, önceki çalışmalardan örnekler ve uygulama alanları tanıtılmaktadır.

PCIBex, tarayıcı tabanlı yapısı sayesinde çevrim içi ortamda başta psikodilbilimsel olmak üzere çok sayıda deneyin yürütülmesine olanak tanır. Yazılı, görsel veya işitsel uyarıların gösterimi ve katılımcı tepkilerinin (tuş basışı, süre, seçim vb.) otomatik kaydını desteklediđi için de farklı alanlarda deneysel testlerin kolayca uygulanmasını mümkün kılar. Alanyazında en çok kullanılan deney türleri ařađıda alt başlıklar halinde anlatılmaktadır.

3.1 Öz-Denetimli okuma testleri

Katılımcı, deney sırasında sözcükleri *tek tek*, öbekleri ya da tümcecikleri *parça parça* ya da cümleleri *bir bütün olarak* okur ve her bir tuş basışı (genellikle boşluk tuşu olarak atanır) arasındaki süre otomatik olarak kaydedilir. Bu süreler, katılımcının okuma hızı ve işleme sürecine ilişkin temel ölçütleri oluşturur. Bu tür bir sunum tekniđi, PennController’ın DashedSentence modülü kullanılarak uygulanır ve çevrim içi okuma deneylerinde yaygın şekilde tercih edilir.

Bunun yanı sıra, katılımcıların okudukları ifadelerle ilgili anlama düzeylerini veya karar verme süreçlerini ölçmek amacıyla farklı modüller de kullanılabilir. Örneđin, `newText()` ve `newScale()` modülleri aracılıđıyla doğru–yanlıř gibi iki seçenekli yargı soruları oluşturulabilir; katılımcıların F veya J gibi belirli tuşlarla yanıt vermesi gereken testler ise `newKey()` ya da `newController("Question")` modülüyle düzenlenir. Bu araçlar, arařtırmacıların hem dil işleme süreçlerini hem de yargı hızlarını aynı deney içinde bütüncül olarak ölçmelerine olanak tanır.

Bu tür zamana duyarlı testler PCIBex’te en sık kullanılan deney araçlarından biridir ve bugüne kadar çok sayıda çalışma bu platform üzerinden yürütülmüştür. Özellikle son yıllarda çevrimiçi veri toplama yöntemlerinin yaygınlařmasıyla birlikte PCIBex kullanılarak gerçekleştirilen arařtırmaların sayısı önemli ölçüde artmıřtır. Bu arařtırmalara Kim & Kim (2025); D’Alesio vd. (2025); Kim & Hwang (2025); Cairncross vd. (2024); Park & Lee (2024); Britton vd. (2024); Lu (2024); Yamaguchi & Ohta (2023); Moulton vd. (2022); Çakır & Çınar (under review); Çınar (2024); Uygun (2022); Gračanin-Yuksekd vd. (2020); Gračanin-Yuksekd vd. (2017) gibi çalışmalar örnek olarak verilebilir.

3.2 Kabul edilebilirlik yargı testleri

PCIBex’te kullanılan bir diđer veri toplama yöntemi kabul edilebilirlik yargı testleridir. Kabul edilebilirlik yargısı testleri, laboratuvar ortamına ihtiyaç duyulmadan uygulanabilen; hatta basit bir kağıt-kalem formatında dahi yürütülebilen yöntemlerdir. Bununla birlikte PCIBex platformu, bu tür testlerin çevrimiçi, kontrollü ve katılımcı dostu bir biçimde uygulanmasını önemli ölçüde kolaylařtırmaktadır. Bu testlerde katılımcılardan, verilen bir cümlenin dilbilgisel ya da anlamsal açıdan ne kadar doğal olduğunu deęerlendirmeleri beklenir. Deęerlendirme genellikle iki tür yanıt formatından biriyle yapılır:

- (i) Likert ölçeęi (örneğin 1–7: “1 = hiç kabul edilebilir deęil” → “7 = tamamen kabul edilebilir”),
- (ii) İkili karar (binary judgement): “Kabul edilebilir / Kabul edilemez”.

PCIBex’te ikili karar temelli kabul edilebilirlik yargıları, farklı modüller aracılığıyla yapılandırılabilir. Bu tür görevler `newScale()` kullanılarak görsel seçenekler üzerinden sunulabileceęi gibi, katılımcıların klavyedeki belirli tuřlara (ör. F = kabul edilebilir, J = kabul edilemez) basarak yanıt verdikleri biçimde de tasarlanabilir. Bu ikinci yaklaşımda, `newKey()` ya da `newController("Question")` denetleyicisi kullanılarak yanıtlar otomatik olarak kaydedilmektedir.

Kabul edilebilirlik yargı testleri oluşturulurken `Template()` işlevi, cümleler, koşullar ve diđer deneysel bilgilerin bulunduğu .csv dosyasındaki her bir maddeyi okuyarak deneme setinin temelini oluşturur. Ardından `newTrial()`, her bir yargı denemesini tanımlar ve deney akışının yapısını belirler. Katılımcıya gösterilecek cümleler ile gerekiyorsa kısa yönergeler `newText()` komutu aracılığıyla ekrana yansıtılır; yargılamanın kendisi ise az önce bahsedildięi gibi `newScale()/newKey()` ya da `newController("Question")` denetleyicisi aracılığıyla gerçekleştirilir.

Alan yazın incelendiğinde, kabul edilebilirlik yargı testlerinin çoęu zaman öz-denetimli okuma gibi başka deney türleriyle birlikte kullanıldığı görülmektedir. Yukarıda deęinilen öz-denetimli okuma çalışmaları arasından Lu (2024), Park & Lee (2024), D’Alesio vd. (2025), Moulton vd. (2022) ve Çınar (2024), bu yöntemin farklı deneysel paradigmalarda bir arada uygulanmasına örnek teşkil etmektedir.

3.3 Dinleme/İřitsel Uyarı Temelli Testler

PCIBex platformunda işitsel testler, katılımcılara ses uyarınları—örneğin sözcükler, cümleler veya en küçük çiftler—sunarak onların karar, algı ya da yargı temelli tepkilerini toplamaya dayanır. Bu tür testler özellikle sesbilgisi, sesbilim, işitsel algı ve cümle işleme alanlarında yaygın biçimde kullanılmaktadır. Kabul edilebilirlik yargı testlerinde olduęu gibi, katılımcılardan dinledikleri ses uyarısına ilişkin doğru/yanlıř ya da uygun/uygun deęil türü kararlar vermeleri istenebilir.

İřitsel testlerin genel işleyiři birkaç temel aşamadan oluşur. Öncelikle, ses dosyaların olarak uyarınlara sisteme yüklenir. Ardından, katılımcı-

cıya sunulan ses uyaranı—örneğin tek bir sözcük, bir cümle veya daha uzun bir ifade—çalıřtırılır ve katılımcı bu uyaranı dinler. Sonrasında, katılımcının tepkisini kaydetmek amacıyla bir düğmeye tıklaması, bir tuřa basması ya da bir ölçek üzerinden seçim yapması istenir. Son ařamada ise sistem, verilen yanıtı ve tepki süresini otomatik olarak kaydeder.

Bu yapıyı uygulamak için `PennController`'ın `newAudio()` modülü ses dosyalarının (.wav veya .mp3 formatında) çalınmasını saęlar. Katılımcı tepkilerini toplamak için `newButton()` veya `newKey()` kullanılabilir; dereceli ya da kategorik deęerlendirmelerin gerektięi durumlarda ise `newScale()` modülü tercih edilir. Bu tür iřitsel karar görevlerinin kullanımına Plaut-Forckosh & Meir (2025) ile Ruda & Huang (2025) çalıřmalarında örnekler bulunmaktadır.

3.4 Sözcüksel Karar Verme Testleri

Sözcüksel karar verme testleri (lexical decision tasks), katılımcılardan dinledikleri ya da gördükleri bir birimin—örneğin bir harf dizisi ya da ses dizisi—gerçek bir sözcük olup olmadıęını mümkün olan en kısa sürede belirlemelerini ister. Bu testlerin temel amacı, sözcük tanıma sürecinin hızını ve biliřsel yükünü ölçmektir. Böylece řu tür soruların yanıtlanmasına katkı saęlar: Katılımcılar gerçek sözcükleri ne kadar hızlı tanır? Sözcük sıklıęı, uzunluęu ve biçimbilimsel yapısı tanıma süresini nasıl etkiler? Hangi tür sözcüksel özellikler tanımayı kolaylařtırır ya da zorlařtırır?

PCIBex platformunda sözcüksel karar verme görevleri, görsel uyarıların sunumu için `newText()`, iřitsel uyarıların sunumu için `newAudio()` modüllerini kullanarak yapılandırılır. Katılımcıların verdikleri yanıtları toplamak amacıyla `newKey()` veya `newButton()` komutları kullanılır. Her bir uyarının ekranda ne kadar süreyle kalacaęını belirlemek ve deney akıřını zamanlamak için ise `newTimer()` iřlevinden yararlanılır. Bu yapı sayesinde deneysel kořullar arasında tutarlı, ölçülebilir ve karřılařtırılabilir tepki süreleri elde edilir.

Sözcüksel karar verme paradıęması, psikodilbilim alanında yaygın biçimde uygulanmakta olup farklı dillerde, yazı sistemlerinde ve biliřsel süreçlerde sözcük tanımanın doęasını inceleyen pek çok çalıřma tarafından kullanılmıřtır. Bu çalıřmalara Lee & Kaiser (2021); Hatzidaki & Santesteban (2024); Zhang vd. (2023); Liu vd. (2025); Ariga & Hirose (2025) gibi arařtırmalar örnek olarak verilebilir.

3.5 Göz İzleme Testleri

PCIBex, donanım tabanlı bir göz izleme (eye-tracking) sistemi sunmamakla birlikte, çevrim içi ortamlarda sıklıkla kullanılan göz-hareketi benzeri ölçütlerin elde edilmesine imkan veren çeřitli araçlar ve modüller saęlamaktadır. Bu yaklařımların amacı, katılımcının görsel bir alanı nasıl taradıęını, hangi bölgelere ne kadar süre odaklandıęını veya belirli bir uyarıyı ne zaman hedefledięini dolaylı yollarla ölçmektir. Böylece klasik göz izleme çalıřmalarında kullanılan ilk bakıř süresi, toplam bakıř süresi veya yeniden

yönelme gibi ölçümlerin çevrimiçi koşullar altında yaklaşık olarak modellenmesi mümkün hale gelir.

PCIBex bu tür görevleri üç farklı teknik üzerinden desteklemektedir:

- (i) görsel bölge tabanlı seçim ve süre ölçümleri,
- (ii) fare (mouse) hareketine dayalı dikkat benzeri izleme,
- (iii) WebGazer.js aracılığıyla webcam-tabanlı bakış tahmini.

Görsel bölge tabanlı uygulamalarda, ekranda yer alacak bölgeler `newImage()`, `newCanvas()` ve `newSelector()` komutlarıyla yapılandırılır; bu sayede nesnelerin konumları hassas biçimde ayarlanabilir. Katılımcının hangi bölgeyi ne zaman seçtiğini belirlemek için `newSelector()` sıklıkla kullanılır; çünkü her seçim zaman damgasıyla birlikte otomatik olarak kaydedilir. Fare hareketinin dikkat yönelimini dolaylı biçimde yansıtabildiği durumlarda `newMouseTracker()` modülü kullanılarak katılımcının imleç hareketleri zaman içinde izlenebilir ve bakış benzeri bir veri elde edilebilir.

Zamanlama gerektiren deneylerde `newTimer()` modülü, uyarıların ekranda ne kadar süreyle kalacağını ve deney akışının hangi hızda ilerleyeceğini kontrol eder. Bu sayede bir bölgeye yönelme hızı, bölgeler arası geçiş süreleri veya toplam bakış benzeri süreler hesaplanabilir.

PCIBex ayrıca, çoklu görsel nesnelere bir sesli uyarı içeren görsel dünya (visual-world) türü deneyleri de destekler. Bu tür bir kurulumda görseller `newCanvas()` ile ekrana yerleştirilir, uyarı `newAudio()` ile sunulur ve katılımcının hangi nesneye yöneldiği, hangi nesneyi seçtiği ya da fare imlecini ilk olarak hangi bölgeye götürdüğü kaydedilir. Böylece çevrim içi bir görsel dünya analizi yapılabilir.

Son olarak, PCIBex'in WebGazer.js ile uyumu sayesinde webcam tabanlı deneyler de yürütülebilmektedir. Bu yöntem, tarayıcı içi kalibrasyonla çalışır ve katılımcının kısa süreli bakış yönelimlerini izlemeye olanak sağlar. Bu yöntemlerin kullanımına Slim vd. (2023) ve Ovans (2025) çalışmalardan örnek gösterilebilir.

Bununla birlikte, PCIBex üzerinde webcam-tabanlı göz izleme kullanımının belirli sınırlılıkları bulunmaktadır. Her şeyden önce, WebGazer.js gibi tarayıcı tabanlı izleme teknolojileri, laboratuvar ortamında kullanılan Tobii veya EyeLink gibi yüksek örnekleme hızına sahip aygıtlara kıyasla hem zaman çözünürlüğü hem de konumsal hassasiyet açısından daha düşük performans göstermektedir. Özsoy vd. (2023), PCIBex/WebGazer ile yürütülen çevrim içi deneylerle karşılaştırıldığında laboratuvar ortamında kullanılan Tobii sisteminin daha güvenilir sonuçlar ürettiğini göstermiştir. Benzer biçimde Patterson vd. (2025), webcam tabanlı göz izleme kullanımının doğal büyüklükte metin okuma gibi göz hareketi analizi gerektiren deneylerde uygun olmadığını belirtmektedir. Ek olarak, webcam-tabanlı izlemenin başarılı olabilmesi katılımcının donanım özelliklerine, ekran çözünürlüğüne, aydınlatma koşullarına ve kalibrasyon doğruluğuna büyük ölçüde bağlıdır. Patterson vd. alandaki çalışmaların önemli bir kısmında minimum donanım gereksinimleri ile kalibrasyon prosedürlerinin yeterli ayrıntıyla rapor-

lanmadığını, bunun da deneylerin tekrarlanabilirliğini zayıflattığını vurgulamaktadır.

4. Örnek bir Deney Tasarımı ve Uygulama Süreci: Öz-Denetimli Okuma Testi

Bu bölümde, PennController eklentisi kullanılarak geliştirilen çevrim içi bir öz-denetimli okuma deneyinin tasarım ve uygulama süreci ayrıntılı biçimde sunulmaktadır. Deney, katılımcı etkileşimini aşamalı olarak yöneten ve belirli görevleri yerine getiren bir dizi modülden oluşmaktadır. Aşağıda, deneyin temel bileşenleri ve bu bileşenlerin işlevleri sırasıyla açıklanmaktadır. Deneyin çevrim içi uygulamasına ise şu adresten erişilebilir: <https://farm.pcibex.net/p/fMvTSf/>

Deneye ait .csv veri dosyaları, deney kodları ve ilgili materyaller ise Açık Bilim Çerçevesi (OSF) üzerinden şu adreste paylaşılmaktadır: https://osf.io/uyq82/overview?view_only=a41eca13771149e39be8260f8d7cb24

4.1 Çalışma içeriği ve veri hazırlama

Örnek verilecek bu çalışma, Türkçede içe yerleşik tümcelerde boş ve açık özne adıllarının ana tümcedeki öncüllere (AÖ Özne, ne-sorusu özne, niceleyici özne) gönderiminin nasıl işlemlendiğini, bağlamlı cümleler aracılığıyla incelemektedir. Deney kapsamında katılımcılara her denemede önce kısa bir bağlam sunulmakta, ardından bu bağlamla ilişkili hedef cümle ekrana getirilmektedir. Hedef cümle okunduktan sonra ise katılımcılar adılın gönderimine yönelik Evet ya da Hayır tuşlarından birini seçmektedir. Örnek bir deneme aşağıda verilmiştir:

- (1) **Bağlam:**
Tüm öğrenciler bilgi yarışmasına çok iyi hazırlandılar.
Hedef cümle:
Onlar birinci olacaklarını düşünüyorlar.
Evet (Özne kendisi hakkında konuşuyor)
Hayır (Özne başkası hakkında konuşuyor)

Katılımcılar, bağlamı okuduktan sonra hedef cümleyi sözcük sözcük ilerleyen bir öz-denetimli okuma düzeninde okumaktadır. Çalışmanın temel amacı, hedef cümlede yer alan kritik sözcüklerin (boş ya da açık adıldan sonraki ilk iki sözcük) işlemlenmesine ilişkin tepki sürelerinin bağlama bağlı olarak nasıl farklılaştığını incelemektir. Bu doğrultuda, katılımcıların kritik sözcükte ve izleyen bölgede gösterdikleri okuma süreleri karşılaştırmalı olarak çözümlenmektedir.

Bu aşamada, deneyde kullanılacak bağlam metinleri ve hedef cümlelerin, bunlara karşılık gelen deneysel tür etiketleriyle birlikte bir .csv dosyasında tanımlanması gerekmektedir. Çalışma kapsamında hazırlanan bu dosya, virgülle ayrılmış sütunlardan oluşmakta olup her bir satır bir deney maddesine karşılık gelmektedir. İlgili .csv dosyasında yer alan temel sütunlar sırasıyla group (deneysel grup), item (soru/uyaran numarası), context (bağlam

metni), sentence (hedef cümle) ve type (soru/koşul türü) biçiminde yapılandırılmıştır. Bu düzenleme, deney sırasında her bir uyarının sistematik biçimde sunulmasını ve sonuç dosyasında koşul-temelli çözümlerinin tutarlı biçimde yürütülmesini sağlamaktadır.

4.2 Bilgilendirme ve gönüllü katılım onamı

Katılımcı içeren deneysel çalışmalarda etik kurul onayının alınması ve katılımcıların arařtırmaya gönüllü olarak katıldıklarını açık biçimde beyan etmeleri zorunludur. Bu gereklilik, çevrim içi ortamda yürütülen deneyler için de geçerlidir. Bu nedenle, katılımcı onamının veri toplama süreci başlamadan önce, çalışmanın en başında alınması gerekmektedir.

Bu çalışma kapsamında deney, katılımcılara arařtırmanın amacı, kapsamı ve etik ilkeleri hakkında bilgi veren bir gönüllü katılım onam sayfası ile başlatılmıştır. Söz konusu bilgilendirme aşamasında, çalışmanın akademik bir arařtırma olduđu, katılımcılardan toplanan verilerin anonim biçimde saklanacağı ve bu verilerin yalnızca bilimsel amaçlarla kullanılacağı açıkça belirtilmiştir. Katılımcıların deneye devam edebilmeleri için, onamlarını açık ve bilinçli bir şekilde vermeleri zorunlu tutulmuştur.

Bilgilendirme ve onam sürecine ilişkin kullanılan örnek kod aşağıda sunulmaktadır. Verilen örnek kodlarda çalışmanın içeriđi daha kısa verilmiş olup, ayrıntılı açıklamalar OSF üzerinde paylaşılan deney dosyalarında yer almaktadır.

```
newTrial("notice",
  newText("consentText", `
    <h2 style="text-align:center;">Gönüllü Katılım Onay Formu</h2>

    <p><b>"The Effect of Antecedents in Processing of Pronominal Binding in
    Turkish"</b>
    başlıklı çalışma, Necmettin Erbakan Üniversitesi ve İstanbul Medeniyet
    Üniversitesi akademik
    personellerinin yürüttüğü bir arařtırmadır.</p>

    <ol>
    <li>Çalışmanın amacı ve süresi hakkında bilgilendirildim.</li>
    <li>Katılımımın gönüllü olduğunu ve dilediğim zaman çekilebileceğimi
    biliyorum.</li>
    </ol>
  `).print(),

  newButton("consentBtn", "ÇALIŞMAYA KATILMAYA ONAY VERİYORUM")
    .print()
    .wait()
);
```

Kullanılan modüllerin işlevleri aşağıda açıklanmaktadır:⁴

```
newTrial("notice", ...)
```

⁴ Deney kodunda yer alan ve modüler yapıların dışında kalan bazı bölümler, metnin görsel sunumuna ilişkin teknik ayarları içermektedir. Bu ayarlar; metinlerin punto büyüklüğü, yazı tipi, hizalama, kenar boşlukları ve benzeri biçimsel özellikleri düzenlemeye yöneliktir. Söz konusu düzenlemeler, deneyin kuramsal yapısı ya da işlevsel akışıyla doğrudan ilişkili değildir. Bu nedenle, çalışmanın yönetsel açıklaması kapsamında yalnızca deneyin işleyişini ve bilişsel süreçleri etkileyen modüller ele alınmış; biçimsel sunuma ilişkin bu tür teknik ayrıntılar ayrıntılı olarak tartışılmamıştır. İlgili ayarlar, genel JavaScript kullanımında yaygın olarak bilinen ve PennController dokümantasyonunda standart biçimde sunulan uygulamalara dayanmaktadır.

Bu komut bir deneme/sayfa oluřturmaktadır. Söz konusu sayfa, deneyin ilk ařaması olarak yapılandırılmıř olup yalnızca bilgilendirme ve gönüllü katılım onamı sürecine ayrılmıřtır.

```
newText("consentText", ...)
```

Katılımcılara sunulan onam metninin ekrana yazdırılması amacıyla kullanılmıřtır. Çalışmanın amacı, gizlilik ilkeleri, gönüllülük esasları ve katılımcı haklarına iliřkin tüm açıklamalar bu modül içerisinde tanımlanmıřtır.

```
newButton("consentBtn", "...")
```

Katılımcının çalışmaya katılmayı kabul ettiđini belirtmesi için kullanılan onay butonunu ve ilgili metni oluřturmaktadır. Bu buton, katılımcının bilinçli onam verdiđini açık biçimde iřaretlemesini sađlamaktadır.

```
.print()
```

Tanımlanan onay butonunun ekranda görünür hale getirilmesini sađlamaktadır.

```
.wait()
```

Katılımcı ilgili butona tıklayana kadar sayfayı bekleme durumunda tutmaktadır. Bu iřlev sayesinde, katılımcının açık onam vermeden deneyin bir sonraki ařamasına geçiři mümkün olmamaktadır.

4.3 Yönergeler

Gönüllü katılım onam sayfasını takiben, deneyin nasıl yürütüleceđini ayrıntılı biçimde açıklayan bir yönerge ekranı katılımcılara sunulmuřtur. Bu bölümde katılımcılara, deneyin yalnızca bilgisayar üzerinden gerçekteřtirileceđi; deney sırasında cümlelerin maskeleyme (moving-window technique; Just vd. 1982) yöntemiyle sunulacađı; ilerlemek için her ařamada boşluk (space) tuşunun kullanılacađı ve okuma tamamlandıktan sonra Evet (F tuşu ile)/Hayır (J tuşu ile) seçeñeklerinden birinin seçilmesinin beklendiđi açıkça belirtilmiřtir. Yönergeler, örnek bađlamalar ve örnek cümleler üzerinden ayrıntılı biçimde açıklanarak katılımcıların deneyi dođru şekilde anlamaları sađlanmıřtır. Yönerge sayfası oluřturmak için kullanılan örnek kod ařađıda sunulmuřtur:

```
newTrial("intro",
  newText("instructions", `
    <h2 style="text-align:center;">Teste Bařlamadan Önce Lütfen Okuyunuz</h2>

    <p>Bu çalışma yalnızca <b>bilgisayar</b> üzerinden yürütölmektedir.</p>

    <p>Cümleler <b>maskeleyme (moving-window)</b> yöntemiyle sunulacaktır.
    İlerlemek için her zaman <kbd>BOŞLUK</kbd> (Space) tuşuna basınız.</p>

    <p>Cümle tamamlandıđında:</p>
    <ul>
      <li><b>Evet</b> için <kbd>F</kbd> tuşuna basınız.</li>
      <li><b>Hayır</b> için <kbd>J</kbd> tuşuna basınız.</li>
    </ul>

    <p>Lütfen cümleleri normal hızınızda okuyunuz ve testi ara vermeden
    tamamlayınız.</p>
  `).print(),
  newButton("continue", "Devam Et")
    .print())
```

```
.wait()  
);
```

Yönerge ekranının oluşturulmasında kullanılan modüller ve işlevleri ise aşağıda sunulmaktadır:

```
newTrial("intro", ...)
```

Yönergeler için deney akışı içerisinde ayrı bir sayfa/deneme oluşturmaktadır. Bu deneme, onam sayfasının hemen ardından yer almakta olup herhangi bir deneysel veri toplamamaktadır.

```
newText("instructions", ...)
```

Yönergelerin tamamının ekrana yazdırılması amacıyla kullanılmaktadır. Görev tanımı, okuma biçimi ve tuşların kullanımıyla ilgili tüm açıklamalar bu modül içerisinde tanımlanmaktadır. Sunulan metin yalnızca ekranda görüntülenmekte olup sonuç dosyasına kaydedilmemektedir.

```
newButton("continue", ...)
```

Katılımcının yönergeleri okuduğunu ve deneye devam etmeye hazır olduğunu belirtmesini sağlayan butonu oluşturmaktadır.

```
.print()
```

Tanımlanan butunun ekranda görünür hale getirilmesini sağlamaktadır.

```
.wait()
```

Katılımcı butona tıklayana kadar sayfayı bekleme durumunda tutmaktadır. Bu işlev sayesinde katılımcıların bir sonraki aşamaya ancak yönergeleri onayladıktan sonra geçmeleri sağlanmaktadır.

4.4 Katılımcı bilgi formu

Yönergelerin ardından katılımcılardan kısa bir katılımcı bilgi formu doldurmaları istenmiştir. Bu form aracılığıyla katılımcıların ad-soyad, yaş ve ana dili demografik bilgileri toplanmıştır.

Girilen bilgiler, deney süresince her denemeye ait verilerle ilişkilendirilmek üzere kaydedilmiştir. Veri girişinin eksiksiz ve tutarlı biçimde sağlanabilmesi amacıyla form alanlarına belirli kısıtlamalar uygulanmıştır. Buna göre, katılımcıların ad ve soyad alanına en az iki sözcükten oluşan metin girmeleri; yaş alanına yalnızca sayısal değer girmeleri; ana dili alanına ise metin girdisi sağlamaları zorunlu tutulmuştur. Örnek kod aşağıda verilmiştir:

```
newTrial("form",  
  newText("t1", "Adınız ve Soyadınız").print(),  
  newTextInput("fullname")  
    .length(64)  
    .print(),  
  
  newText("t2", "Yaşınız").print(),  
  newTextInput("age")  
    .length(4)  
    .print(),  
  
  newText("t3", "Ana Diliniz").print(),  
  newTextInput("language")  
    .length(64)  
    .print(),
```

```
newButton("send", "Onayla ve İlerle")
    .print()
    .wait()
    .test(
        getTextInput("fullname").test.text(/^\\s*[a-zıİöüşçğ]+(\\s+[a-
zıİöüşçğ]+)\\s*$/i)
        .and(getTextInput("age").test.text(/^\\s*\\d+\\s*$/i))
        .and(getTextInput("language").test.text(/^\\s*[a-zıİöüşçğ]+(\\s+[a-
zıİöüşçğ]+)\\s*$/i))
    ),
    newVar("fullname").global().set(getTextInput("fullname")),
    newVar("age").global().set(getTextInput("age")),
    newVar("language").global().set(getTextInput("language"))
);
```

Katılımcı bilgi formunun PennController ortamında çalıştırılmasında aşağıda açıklanan modüller kullanılmıştır:

```
newTrial("form", ...)
```

Demografik bilgi toplamak amacıyla deney akışında ayrı bir sayfa/deneme oluşturmaktadır. Bu deneme, yönergelerden sonra yer almakta ve test aşamasından önce sunulmaktadır.

```
newText (...)
```

Form alanlarına ait başlıkların (Ad-Soyad, Yaş, Ana Dil) ekrana yazdırılmasını sağlamaktadır. Bu metinler yalnızca görsel etiket işlevi görmektedir.

```
newTextInput("fullname" / "age" / "language")
```

Katılımcılardan metin girdisi almak için kullanılmaktadır. Her bir TextInput öğesi, farklı bir demografik değişkene karşılık gelmektedir.

```
.length (...)
```

Girdi alanlarına yazılabilecek maksimum karakter sayısını sınırlandırmaktadır.

```
.print()
```

Tanımlanan öğelerin ekranda görünür hale getirilmesini sağlamaktadır.

```
newButton("send", "Onayla ve İlerle")
```

Katılımcının formu tamamladığını ve bir sonraki aşamaya geçmek istediğini belirtmesini sağlayan “Onayla ve İlerle” butonunu oluşturmaktadır.

```
.wait (...)
```

Butonun yalnızca tüm alanlar geçerli biçimde doldurulduğunda etkinleşmesini sağlamaktadır. Bu amaçla düzenli ifadeler (*regular expressions, regex*) kullanılarak, ad-soyad alanında en az iki sözcük bulunması, yaş alanına sayısal bir değer girilmesi ve ana dili alanına metin girdisi sağlanması zorunlu tutulmuştur.

```
newVar (...).global().set (...)
```

Formdan elde edilen bilgileri global değişkenlere atamaktadır. Bu sayede katılımcı bilgileri, test denemeleri sırasında her satıra eklenerek sonuç dosyasında sistematik biçimde kaydedilmektedir.

4.5 Alıřtırma ařaması

Ana test ařamasına geilmeden nce, katılımcıların teste alıřmalarını saęlamak amacıyla bir alıřtırma blm uygulanmıřtır. Alıřtırma maddeleri deney maddeleriyle aynı yapıya sahip olmakla birlikte `.log()` kullanılmadıęı iin okuma sreleri ve yanıtlar kaydedilmemektedir. Bu ařama, katılımcıların szck szck okuma teknięine ve yanıt verme biimine uyum saęlamalarını amalamaktadır. Alıřtırma ařamasını oluřturmak iin kullanılan rnek kod ařaęıda sunulmuřtur:

```
Template("practice.csv", row =>
  newTrial("practice",
    newText("label", "Alıřtırma").print(),

    newText("context", row.context).print(),
    newNexter("nextContext", getText("context").remove()),

    newController("DashedSentence", { s: row.sentence.trim() })
      .wait()
      .remove(),

    newController("Question", {
      as: [{"f", "Evet (F)"}, {"j", "Hayır (J)"}],
      randomOrder: false,
      presentHorizontally: true
    })
      .wait()
      .remove()
  )
);
```

Kullanılan modller ve iřlevleri ise ařaęıda verilmiřtir:

```
Template("practice.csv", row => ...)
```

Alıřtırma uyarılarını `practice.csv` dosyasından satır satır okumakta ve her satır iin bir deneme retmektedir.

```
newTrial("practice", ...)
  practice etiketli bir alıřtırma denemesi oluřturmaktadır.
```

```
newText("label", ...)
```

Denemenin alıřtırma olduęunu belirten kısa bir etiket/metin sunmaktadır.

```
newText("context", row.context)
```

`.csv`'den gelen baęlam metnini ekranda grntlemektedir.

```
newNexter(..., getText("context").remove())
```

Bořluk tuřu girdisini algılamaktadır. Tuřa basıldıęında baęlam metnini kaldırarak bir sonraki ařamaya geiři saęlamaktadır.

```
newController("DashedSentence", ...)
```

z-denetimli okuma sunumunu gerekleřtirmektedir. Bu modlle katılımcı bořluk tuřuyla szck szck ilerlemektedir. Cmle tamamlanmadan sonraki ařamaya geiře izin verilmez.

```
newController("Question", ...)
```

Okuma tamamlandıktan sonra ikili karar sorusunu (Evet / Hayır) sunmaktadır. Bu noktada F ve J tuřları yanıt seeneklerine atanmıřtır. Yanıt verilmeden deneme sona ermez.

4.6 Test ařaması

Deneyin temel bölümünü oluřturan test ařamasında, deneysel uyaranlar katılımcılara rastgele sırayla sunulmuřtur. Her test denemesi ařağıdaki bileřenlerden oluřmaktadır:

4.6.1 Baėlam sunumu

Her deneme, hedef cümleden önce sunulan kısa bir baėlam metni ile bařlamaktadır. Katılımcı boşluk tuřuna bastıėında baėlam metni ekrandan kaldırılır ve hedef cümleye geilir.

4.6.2 Öz-denetimli okuma (sözcük sözcük sunum)

Hedef cümle, öz-denetimli okuma modülü kullanılarak sözcük sözcük sunulmuřtur. Katılımcı her boşluk tuřuna bastıėında yalnızca bir sözcük görünüp, önceki sözcük maskeleye yöntemiyle gizlenmektedir. Bu yöntem, katılımcıların cümleyi kendi okuma hızlarında iřlemlemelerine olanak tanımaktadır. Okuma süreci boyunca her sözcük için tepki süreleri kaydedilmektedir.

4.6.3 Soru ve yanıt

Hedef cümlelerin tamamlanmasının ardından katılımcılara ikili bir karar sorusu yöneltilmektedir. Katılımcılar, cümlede yer alan öznenin kendisinden mi yoksa bařka bir kiřiden mi bahsettiėine karar vererek: Evet (F tuřu) ya da Hayır (J tuřu) seeneklerinden birini iřaretlemektedir. Verilen yanıtlar ve yanıt süreleri de otomatik olarak kaydedilmektedir. Denemelerde ayrıca, deneysel kořullara baėlı olarak doėru yanıt tanımlamaları yapılmıř ve bu bilgiler analiz sürecinde kullanılmak üzere kaydedilmiřtir. Test ařamasını oluřturmak için kullanılan örnek kod ise ařağıda sunulmuřtur:

```
Template("test.csv", row =>
  newTrial("test",
    // 1) Baėlam
    newText("context", row.context).print(),
    newNexter("nextFromContext", getText("context").remove()),

    // 2) Moving-window öz-denetimli okuma (RT kaydı açık)
    newController("DashedSentence", { s: row.sentence.trim() })
    .log()
    .wait()
    .remove(),

    // 3) İkili karar sorusu (F/J) - yanıt + RT kaydı açık
    newController("Question", {
      as: [{"f", "Evet (F)"}, {"j", "Hayır (J)"}],
      hasCorrect: row.type.trim().endsWith("/null") ? 0 : 1,
      randomOrder: false,
      presentHorizontally: true
    })
    .log()
    .wait()
    .remove()
  )
  // 4) Deneme-etiketleri (kořul/uyaran bilgisi)
```

```
.log("group", row.group)
.log("item", row.item)
.log("type", row.type)

// 5) Katılımcı bilgileri (formdan gelen global deęişkenler)
.log("fullname", getVar("fullname"))
.log("age", getVar("age"))
.log("language", getVar("language"))
);
```

Kullanılan modüllerin ayrıntılı açıklaması ařağıdadır:

```
Template("test.csv", row => ...)
```

Ana test uyarılarını test.csv dosyasından satır satır okumakta ve her satır için bir test denemesi üretmektedir.

```
newTrial("test", ...)
```

Bu modül ana test denemesini oluřturmaktadır. Denemeler tipik olarak `Sequence(..., randomize("test"), ...)` ile rastgele sırada sunulmaktadır.

```
newText("context", row.context).print()
```

Baęlam metnini katılımcılara ekranda sunmaktadır (.csv'den okur).

```
newNexter("...", getText("context").remove())
```

Katılımcının boşluk tuřuna basmasını bekler; basılınca baęlam metnini kaldırarak hedef cümle ařamasına geçiři saęlamaktadır.

```
newController("DashedSentence", ...).log().wait().remove()
```

Öz-denetimli okuma sunumunu boşluk tuřuyla sözcük sözcük gerçekleřtirmektedir.

.log(): sözcük/tuř basışı düzeyinde okuma sürelerini kaydeder.

.wait(): cümle tamamlanmadan sonraki ařamaya geçiře izin vermez.

.remove(): tamamlandıęında öęeyi ekrandan kaldırır.

```
newController("Question", ...).log().wait().remove()
```

Okuma sonrasında ikili karar sorusunu sunmaktadır (F=Evet, J=Hayır).

as: seęenekleri tuřlara baęlar.

presentHorizontally:true seęenekleri yatay; randomOrder:false sıralamayı sabit gösterir.

hasCorrect: doęru yanıt tanımlamaktadır (0=ilk seęenek/F, 1=ikinci seęenek/J).

.log(): yanıt ve yanıt süresini kaydeder.

.wait(): yanıt verilmeden denemeyi ilerletmez.

.remove(): yanıt sonrası soruyu kaldırır.

```
.log("group"/"item"/"type", ...)
```

Denemeyi kořul/uyaran etiketleriyle iřaretlemektedir; sonuç dosyasında filtreleme ve analiz için kullanılmaktadır (bkzn., bölüm 4.7).

```
.log("fullname"/"age"/"language", getVar(...))
```

Katılımcı bilgi formundan gelen global deęişkenleri her denemeye eklemektedir; böylece sonuç dosyasında deneme düzeyinde katılımcı-demografik bilgi eşleřtirmesi yapılabilir.

4.6.4 Deneyin sonlandırılması ve veri kaydı

Tüm test denemeleri tamamlandıktan sonra deney bir kapanıř ekranı ile sonlandırılmıřtır. Deney süresince toplanan okuma süreleri, yanıtlar, ve katılımcı bilgileri sistem tarafından otomatik olarak kaydedilmiř ve analiz için hazır hale getirilmiřtir. Buna iliřkin örnek kod ařaęıda sunulmaktadır:

```
Sequence(  
  "notice",  
  "intro",  
  "form",  
  "practice",  
  randomize("test"),  
  SendResults(),  
  "outro",  
);  
newTrial("outro",  
  newText("outro", "Görüşürüz! See you! Arrivederci! Auf wiedersehen! じゃあ  
ね!"),  
  newButton("wait4me", "").wait(),  
);
```

Kullanılan modül ve işleri ise ařaęıda verilmiřtir:

```
SendResults()
```

Deney boyunca toplanan tüm verilerin (tepki süreleri, yanıtlar, ve katılımcı bilgileri) PCIBex sunucusuna gönderilmesini saęlamaktadır. Bu modül çalıştırılmadan deney sonlandırılırsa veriler kaydedilmez. Bu nedenle, deney akıřında test ařamasından hemen sonra konumlandırılmıřtır.

```
Sequence(...)
```

Deneyin genel akıř sırasını belirlemektedir.

```
newTrial("outro", ...)
```

Deneyin son ekranını (kapanıř sayfasını) oluřturmaktadır.

```
newText("outro", ...)
```

Kapanıř mesajını katılımcılara ekranda sunmaktadır.

```
newButton("wait4me", "").wait()
```

Deneyin otomatik olarak kapanmasını engeller. Katılımcı bu ekranda kalıp tarayıcıyı kendi isteęiyle kapatabilmektedir. Bu yaklařım, SendResults() işleminin tamamlanması için güvenli bir bitiş saęlamaktadır.

4.7 Sonuç Dosyasını Okuma

Bu çalışmada elde edilen deneysel veriler, PennController altyapısı tarafından üretilen olay-temelli (event-based) bir sonuç dosyasında kaydedilmiřtir. Sonuç dosyası, deney sırasında gerçekteşen her etkileşimi ayrı bir satır olarak kaydeden uzun formatlı bir yapıya sahiptir. Ařaęıda, sonuç dosyasında yer alan sütunların nasıl yorumlanacaęı ve bu sütunların deney kodunda tanımlanan ařamalarla nasıl iliřkilendirileceęi, tek bir denemeye ait örnek satırlar üzerinden ayrıntılı biçimde açıklanmaktadır.

Dosyanın en üst kısmında # işaretiyle başlayan satırlar yer almaktadır. Bu satırlar, deneyin çalıştırıldığı tarih, tarayıcı bilgisi ve teknik ayarlara ilişkin meta-bilgiler içermektedir. Bu bilgiler istatistiksel analizlerde kullanılmamış ve veri işleme aşamasında dışlanmıştır. Bu bilgiler aşağıda örneklendirilmiştir:

```
#  
# Results on Tue, 17 Dec 2024 12:37:49 GMT  
# USER AGENT: Mozilla/5.0 (Windows NT 10.0  
# Design number was non-random = 0
```

Analiz edilen kısım ise, bu meta-bilgilerin ardından gelen ve gerçek deney olaylarını içeren satırlardan oluşmaktadır. Sonuç dosyasında yer alan her satır, deney sırasında gerçekleşen tek bir olayı temsil etmekte ve bu olaylara ilişkin bilgiler PennController tarafından önceden tanımlanmış sabit bir sütun sırası içerisinde kaydedilmektedir. Bu sütunlar sırasıyla⁵ Controller name, Order number of item, Inner element number, Label, Latin Square Group, PennElementType, PennElementName, Parameter, Value ve EventTime alanlarını içermektedir. Bu temel teknik sütunları, deney kodunda .log() komutları aracılığıyla eklenen group, item, type, fullname, age ve language sütunları izlemektedir.

Bu sütunların sonuç dosyasındaki görünümü aşağıda sunulmaktadır. Her bir katılımcıya ait bu bilgiler, sonuç dosyasında bir kez yer almaktadır:

```
#  
# Columns below this comment are as follows:  
# 1. Results reception time.  
# 2. MD5 hash of participant's IP address.  
# 3. Controller name.  
# 4. Order number of item.  
# 5. Inner element number.  
# 6. Label.  
# 7. Latin Square Group.  
# 8. PennElementType.  
# 9. PennElementName.  
# 10. Parameter.  
# 11. Value.  
# 12. EventTime.  
# 13. Comments.  
  
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,1,0,notice,NULL,Penn  
Controller,0,_Trial_,Start,1734435806773,NULL
```

⁵ Sonuç dosyasında yer alan sütun adlarının Türkçe karşılıkları aşağıdadır:

Controller name → Denetleyici adı
Order number of item → Denemenin deney içindeki sıra numarası
Inner element number → Deneme içi öge numarası
Label → Deneme etiketi
Latin Square Group → Latin kare grubu
PennElementType → PennController öge türü
PennElementName → PennController öge adı
Parameter → Parametre / olay türü
Value → Değer (uyarıcı ya da yanıt içeriği)
EventTime → Olay zamanı (ms)
group → DeneySEL grup
item → Uyarıcı numarası
type → DeneySEL koşul
fullname → Katılımcı adı-soyadı
age → Yaş
language → Ana dili
Reading time → Okuma süresi
Whether or not answer was correct → Yanıtın doğru olup olmadığı
Time taken to answer → Yanıt verme süresi
DashedSentence → ÖZ-denetimli okuma (maskeleme) sunumu

```
# 13. group.
# 14. item.
# 15. type.
# 16. fullname.
# 17. age.
# 18. language.
# 19. Comments.
```

```
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,12,0,test,NULL,PennC
ontroller,11,_Trial_,Start,1734438850287,1,5,Q-universalNEG /null,AD-SOYAD,19,
Türkce,NULL
```

Öz-denetimli okuma ve karar aşamasına ilişkin ölçümler ise dosyanın devamında Reading time, Whether or not answer was correct ve Time taken to answer sütunlarında yer almaktadır. Her okunan cümle başında bu sütunlar yer almaktadır. Deney kapsamında kaç cümle okunmuşsa, bu sütunlar da her cümle için sonuç dosyasında aynı biçimde yinelenmektedir. Sözcüklerin kaç milisaniye (ms) içinde okunduğuna ilişkin süre bilgileri, öz-denetimli okuma ölçümü kapsamında bu sütunlar aracılığıyla kaydedilmektedir:⁶

```
# 19. Reading time.
# 20. Newline?.
# 21. Sentence (or sentence MD5).
# 22. Comments.
```

```
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,12,0,test,NULL,Contr
oller-DashedSentence,DashedSentence,1,Kimse,1734438868224,1,5,Q-universalNEG
/null, AD-SOYAD,19,Türkce,377,false,Kimse kopya cektigini söylemedi.,Any
additional parameters were appended as additional columns
```

```
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,12,0,test,NULL,Contr
oller-DashedSentence,DashedSentence,2,kopya,1734438868224,1,5,Q-universalNEG
/null, AD-SOYAD,19,Türkce,377,false,Kimse kopya cektigini söylemedi.,Any
additional parameters were appended as additional columns
```

```
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,12,0,test,NULL,Contr
oller-DashedSentence,DashedSentence,3,cektigini,1734438868224,1,5,Q-
universalNEG /null, AD-SOYAD,19,Türkce,401,false,Kimse kopya cektigini
söylemedi.,Any additional parameters were appended as additional columns
```

```
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,12,0,test,NULL,Contr
oller-DashedSentence,DashedSentence,4,söylemedi.,1734438868224,1,5,Q-
universalNEG /null, AD-SOYAD,19,Türkce,2768,false, Kimse kopya cektigini
söylemedi.,Any additional parameters were appended as additional columns
```

Bu sütunların işleyişi, tek bir denemeye ait örnek satırlar üzerinden açıkça görülebilmektedir. Örneğin ana test aşamasında sunulan “Kimse kopya cektigini söylemedi.” cümlesi, sonuç dosyasında PennElementName = DashedSentence olarak etiketlenmiş dört ayrı satırla temsil edilmiştir. Bu satırlarda Controller name alanı olayın Controller-DashedSentence tarafından üretildiğini, Label alanı olayın test denemesine ait olduğunu göstermektedir. Parameter sütunu kelimenin cümle içindeki sırasını (1–4), Value sütunu ise ekranda sunulan kelimeyi (*Kimse*, *kopya*, *cektigini*, *söylemedi*.) içermektedir. Aynı satırlarda yer alan Reading time sütunu, her bir kelimenin ekranda kaldığı süreyi ms cinsinden doğrudan vermektedir.

⁶ Okunabilirliği artırmak amacıyla, örnekte okunan sözcükler ile bu sözcüklerin kaç ms içinde okunduğuna ilişkin süre bilgileri koyu vurgulanmıştır.

Yukarıda verilen örnekte okuma süreleri sırasıyla **Kimse için 337 ms, kopya için 377 ms, çektiğini için 401 ms ve söylemedi. için 2768 ms** olarak kaydedilmiştir. Bu sözcük okuma satırlarının tamamında `group = 1, item = 5` ve `type = Q-universalNEG /null` (öncül: olumsuzluk taşıyan evrensel niceleyici/içe yerleşik tümce öznesi: boş) değerlerinin tekrar etmesi, bu olayların aynı deneysel denemeye ve koşula ait olduğunu açıkça göstermektedir. Aynı şekilde `age` ve `language` sütunlarında yer alan değerler, bu okuma olaylarının ilgili katılımcıyla doğrudan ilişkilendirildiğini ortaya koymaktadır.

Okuma aşamasını takiben sunulan ikili karar sorusuna verilen yanıt, aynı denemeye ait ayrı bir satırla kaydedilmiştir. Bu satırda `PennElementName = Question` olarak belirtilmekte; `Value` sütununda katılımcının verdiği yanıt **Evet (F)** biçiminde açıkça yer almaktadır. Bu yanıt, deney yönergelerinde belirtildiği üzere klavyedeki `F` tuşuna karşılık gelmektedir. Aynı satırda bulunan `Whether or not answer was correct` sütununda değer `1` olarak kaydedilmiş olup, bu durum verilen yanıtın doğru olduğunu göstermektedir. Bu doğruluk bilgisi, deney kodunda tanımlanan `hasCorrect` parametresiyle birebir uyumludur. `Time taken to answer` sütununda yer alan **2008 ms** değeri ise katılımcının soruya yanıt vermek için harcadığı süreyi göstermektedir. Böylece bu deneme için hem sözcük düzeyindeki okuma süreleri hem de karar aşamasına ait yanıt ve yanıt süresi, aynı sonuç dosyası içerisinde açık biçimde izlenebilmektedir.

```
# 19. Whether or not answer was correct (NULL if N/A).
# 20. Time taken to answer..
# 21. Comments.

1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,12,0,test,NULL,Controller-Question,Question,NULL,Evet (F),1734438870233,1,5,Q-universalNEG /null,AD-SOYAD,19, Türkce,1,2008,Any additional parameters were appended as additional columns

# 19. Comments.
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,12,0,test,NULL,PennController,11,_Trial_,End,1734438870234,1,5,Q-universalNEG /null, AD-SOYAD,19,Turkce,NULL
1734439069,8a49962f9d12346acdf3b08b19f23e15,PennController,18,0,test,NULL,PennController,17,_Trial_,Start,1734438870237,1,11,wh-distributive /null, AD-SOYAD,19,Türkce,NULL
```

4.8 Veri çözümleme

Bu bölümde, `PennController` tarafından üretilen sonuç dosyasının R ya da Python gibi programlama tabanlı istatistik ortamına aktarılması ve araştırma sorusuna uygun biçimde çözümlenmesi için izlenen çözümleme akışı sunulmaktadır. Çalışmanın temel hedefi, deneysel koşullara (ör. öncül türü + boş/açık özne adıl türü bilgisi) göre sözcük düzeyinde okuma sürelerinin (ms) ve ikili karar bölümünde yanıt doğruluğu ile yanıt verme sürelerinin (ms) özetlenmesidir.

Analiz süreci üç ana adımdan oluşmaktadır. İlk adımda sonuç dosyası R ya da Python'a aktarılır; PCIBex dosyalarında `#` ile başlayan açıklama satırları veri olmadığı için içe aktarma sırasında dışarıda bırakılmaktadır. İkinci adımda yalnızca ana deney denemeleri seçilmektedir; bunun için `label` değişkeni

“test” olan satırlar filtrelenir. Üçüncü adımda ise veri iki ayrı ölçüm türüne ayrılmaktadır: öz-denetimli okuma verisi ve ikili karar verisi. Öz-denetimli okuma verisi, `penn_element_name` değeri “DashedSentence” olan satırlardan alınır; bu satırlarda her kelime için `Reading time (ms)` doğrudan mevcuttur. Karar verisi ise `penn_element_name` değeri “Question” olan satırlardan elde edilmektedir; burada yanıt (Evet–F / Hayır–J), yanıtın doğruluđu (1/0) ve yanıt verme süresi (ms) ayrı deđişkenler olarak ele alınmaktadır.

Son aşamada, deneysel koşulları temsil eden `type` deđişkenine göre gruplama yapılarak her koşul için ortalama okuma süresi ve standart sapma (SS) hesaplanmıştır. İkili karar aşaması için de aynı şekilde koşul bazında doğruluk oranı ve ortalama yanıt verme süresi üretilmiştir. Böylece, araştırma sorusunun merkezinde yer alan “öncül türüne göre boş/açık özne adlı içeren cümlelerde sözcükler ortalama kaç ms’de okunuyor?” sorusu, koşul bazında özet istatistiklerle doğrudan yanıtlanabilir hale gelmiştir. Aşađıda bu deneyin verilerini çözümleneye ilişkin örnek bir R kodu bulunmaktadır⁷.

```
library(readr)
library(dplyr)
library(stringr)
library(tidyr)

df <- read_csv(
  "results_prod (4).csv",
  comment = "#",
  col_names = FALSE,
  locale = locale(encoding = "UTF-8"),
  show_col_types = FALSE
)

colnames(df) <- c(
  "reception_time", "ip_md5",
  "controller_name", "item_order", "inner_element",
  "label", "latin_square",
  "penn_element_type", "penn_element_name",
  "parameter", "value", "event_time",
  "group", "item", "type", "fullname", "age", "language",
  "col19", "col20", "sentence", "comments"
)

test <- df %>%
  filter(label == "test")

spr <- test %>%
  filter(penn_element_name == "DashedSentence") %>%
  mutate(
    word_pos = as.integer(parameter),
    word = as.character(value),
    reading_time = as.numeric(col19),
    group = as.integer(group),
    item = as.integer(item),
    age = suppressWarnings(as.integer(age))
  )
```

⁷ Sonuçlar Python ortamına da aktarılabilmekle birlikte, bu çalışmada veri çözümlenmesi R ortamı üzerinden anlatılmaktadır.

```

qa <- test %>%
  filter(penn_element_name == "Question") %>%
  transmute(
    item_order = as.integer(item_order),
    group = as.integer(group),
    item = as.integer(item),
    type = as.character(type),
    age = suppressWarnings(as.integer(age)),
    language = as.character(language),
    answer = as.character(value),
    correct = as.numeric(col19),
    answer_time = as.numeric(col20)
  )

spr_clean <- spr %>%
  filter(!is.na(reading_time)) %>%
  filter(reading_time >= 100, reading_time <= 5000)

qa_clean <- qa %>%
  filter(!is.na(answer_time)) %>%
  filter(answer_time >= 200, answer_time <= 10000)

spr_summary <- spr_clean %>%
  group_by(type) %>%
  summarise(
    mean_rt = mean(reading_time),
    sd_rt = sd(reading_time),
    n = n(),
    .groups = "drop"
  )

qa_summary <- qa_clean %>%
  group_by(type) %>%
  summarise(
    acc = mean(correct, na.rm = TRUE),
    mean_answer_time = mean(answer_time),
    sd_answer_time = sd(answer_time),
    n = n(),
    .groups = "drop"
  )

spr_summary
qa_summary

```

Bu kod R programında çalıştırıldığında ortalama okuma süreleri aşağıdaki gibi bir sonuç üretmektedir:

Tablo 1. Sözcük düzeyi ortalama okuma süreleri

type ⁸	Ortalama okuma süresi (ms)	SS
DP/null	792.43	1315.22
DP/overt	1013.17	2448.52

⁸ Bu kısımda öncül türü ve adıl türleri verilmektedir. Örneğin DP/null etiketinde DP, ana tümcedeki öncülün (öznenin) türünü; null ise içe yerleşik tümcede öznenin boş olarak gerçekleştiğini göstermektedir.

type ⁸	Ortalama okuma süresi (ms)	SS
Q-distributive /null	615.13	560.63
Q-distributive /overt	757.29	1376.08
Q-distributiveNEG /null	809.92	1386.99
Q-distributiveNEG /overt	838.85	1537.44
Q-universal /null	694.69	708.47
Q-universal /overt	807.43	1206.74
Q-universalNEG /null	980.29	1479.19
Q-universalNEG /overt	801.77	1271.14
wh /null	985.55	1490.49
wh /overt	858.47	1460.43
wh-distributive /null	888.08	1904.94
wh-distributive /overt	924.65	1576.15

R'dan elde edilen ikili karar sorularının doęruluk oranları ve ortalama yanıt süreleri ise ařaęıdaki tabloda sunulmaktadır:

Tablo 2. Doęruluk ve yanıt süreleri

type	Doęruluk (oran) ⁹	Ortalama yanıt süresi (ms)	SS
DP/null	0.9211	1322.95	868.78
DP/overt	0.3421	2376.45	3579.29
Q-distributive /null	0.8421	2307.08	2650.14
Q-distributive /overt	0.7368	2953.84	4265.46
Q-distributiveNEG /null	0.7895	2236.47	2640.54
Q-distributiveNEG /overt	0.6579	1921.79	1913.04
Q-universal /null	0.7895	2000.71	2197.96
Q-universal /overt	0.6842	2265.45	2795.61
Q-universalNEG /null	0.7632	3515.71	5687.32
Q-universalNEG /overt	0.7368	3168.97	4963.12
wh /null	0.5789	2136.79	2010.94
wh /overt	0.7632	2392.16	2840.21
wh-distributive /null	0.6053	2670.58	3241.79
wh-distributive /overt	0.7105	2462.13	1936.01

Bu noktada vurgulanması gereken önemli bir husus, çalışmada kullanılan okuma sürelerinin (ms) ve yanıt sürelerinin (ms) ölçüm olarak zaten PennController tarafından sonuç dosyasına kaydedildięidir. Dolayısıyla R

⁹ 1=doęru, 0=ise yanlış olarak alınmıştır.

ortamında yapılan iřlem, yeni bir tepki süresi üretmek deęil, sonuç dosyasında mevcut olan bu süreleri deneysel kořullara göre filtreleyerek basit ve sade biçimde özetlemek (ör. kořul bazında ortalama, standart sapma) řeklinde gerçekteřmiştir. Bu bağlamda, R programı burada temel olarak veriyi düzenleme ve tanımlayıcı istatistikleri raporlama iřlevi görmektedir.

Elde edilen kořul bazlı ortalama okuma süreleri ve karar verileri, çözümlene sürecinin ilk ve betimleyici ařamasını oluřturmaktadır. Bu özet çıktıları alındıktan sonra, deneysel kořullar arasındaki farkların çıkarımsal olarak deęerlendirilmesi amacıyla ileri düzey istatistiksel analizlere geçilecektir. Arařtırma sorusunun yapısına baęlı olarak, iki kořulun karřılařtırılması gereken durumlarda t-testi gibi parametrik testler uygulanabilir; ancak öz-denetimli okuma verilerinin katılımcı ve madde düzeyinde tekrarlı ölçümler içermesi nedeniyle, katılımcı ve madde deęiřkenlerini aynı anda modelleyebilen doęrusal karma etkili modeller (linear mixed-effects models) çoęu durumda daha uygun bir çözümlene yaklařımı sunmaktadır. Bu nedenle çalıřmanın sonraki ařamalarında, okuma süreleri ve karar verileri üzerinde uygun varsayımlar gözetilerek hem temel karřılařtırmaların hem de karma etkili model temelli çözümlenelerin yürütülmesi planlanabilir.

6. Sonuç ve Deęerlendirme

Bu çalıřmada, PCIBex platformu ve PennController eklentisinin deneysel dilbilim arařtırmalarında sunduęu olanaklar, yürütülebilecek uygulamalar ve örnek bir öz-denetimli okuma testi tasarımı hem kuramsal hem de uygulamalı düzeyde tartıřılmıştır. Son yıllarda psikodilbilimsel yöntemlerin yaygınlařması ve çevrim içi veri toplama pratiklerinin arařtırmalar için son derece yaygın olması tarayıcı tabanlı deney altyapılarını yalnızca bir seęenek olmaktan çıkarıp giderek daha merkezi bir konuma tařımıştır. Bu bağlamda PCIBex erişilebilir bir ortamda, deney akıřını bařtan sona yönetebilmesi bakımından arařtırmacılar için pratik bir çözümlene sunmaktadır.

Özellikle PennController'ın JavaScript tabanlı, ancak arařtırmacının hızlıca uygulamaya geçebilmesine olanak saęlayan görece sade söz dizimi, psikodilbilimsel paradigmalarda web ortamına aktarılmasında önemli bir avantajdır. Örnek uygulamada da görüldüęü üzere, okuma süresi gibi zamana duyarlı ölçümler ile karar yanıtları aynı denemede birleřtirilebilmekte; deneysel kořullara iliřkin etiketler (group, item, type) ile katılımcı bilgileri (age, language vb.) `.log()` komutları aracılıęıyla her olay satırına sistematik biçimde iliřkilendirilebilmektedir.

Bununla birlikte, çevrim içi deneylerin doęası gereęi bazı yöntemsel sınırlılıklar kaçınılmazdır ve sonuçların yorumlanmasında bu sınırlılıkların açıkça dikkate alınması gerekir. Katılımcıların ekran boyutu, klavye düzeni, tarayıcı performansı, dikkat düzeyi ve ortam gürültüsü gibi arařtırmacının doğrudan kontrol edemedięi deęiřkenler, özellikle tepki süresi ölçümlerinde güçlükler yaratabilir. Bu nedenle PCIBex üzerinden yürütülen zamana duyarlı çalıřmalarda, veri toplama öncesinde minimum teknik gereksinimlerin (ör. yalnızca bilgisayar, belirli tarayıcılar, tam ekran tercihleri)

yönergelerde net biçimde belirtilmesi; veri sonrası aşamada ise dışlama ve temizleme adımlarının (aşırı kısa/uzun süreler, eksik yanıtlar) raporlanması önemlidir. Diğer bir deyişle, PCIBex çevrim içi çalışmayı kolaylaştırırken, arařtırmanın ölçüm kalitesini korumak için tasarım ve çözümleme düzeyinde daha sistemli önlemler alınması da gerektirebilir.

Sonuç olarak bu çalışma PCIBex'in, deney tasarımı ve veri yönetimi açısından arařtırmacıya sağladığı pratik avantajları örnek bir öz-denetimli okuma senaryosu üzerinden görünür kılmış; aynı zamanda çevrim içi veri toplamanın taşıdığı sınırlılıkları da ortaya koymuştur. Bu çerçevede PCIBex, hem psikodilbilimsel arařtırmaların ölçęini büyütme hem de farklı katılımcı profillerine erişimi artırma potansiyeli taşımaktadır.

Teşekkür

Bu çalışmada kullanılan deney kodlarının hazırlanması ve uygulanmasındaki katkılarından dolayı İstanbul Medeniyet Üniversitesi Dilbilimi Lisans öğrencisi Deniz Engin'e teşekkür ederiz.

Kaynakça

- Ariga, T., & Hirose, Y. (2025). Recognition of spoken words with mispronounced lexical prosody in Japanese. *The Journal of the Acoustical Society of America*, 157(6), 4102–4118.
- Britton, J., Cong, Y., Hsu, Y. Y., Chersoni, E., & Blache, P. (2024). On the influence of discourse connectives on the predictions of humans and language models. *Frontiers in Human Neuroscience*, 18, 1363120.
- Cairncross, A., Vogelzang, M., & Tsimpli, I. (2024). Evaluating the pseudorelative-first hypothesis: Evidence from self-paced reading and persistence effects. *Glossa Psycholinguistics*, 3(1).
- Çakır, S., & Çınar, O. (under review). The contextual influences on the pronominal binding preferences in Turkish: A processing-based study.
- Çınar, O. (2024). Revisiting native grammar through L2 theories: Knowledge and processing of null and overt subject pronouns in Turkish. *Nesne Psikoloji Dergisi*, 12(33), 331–350.
- D'Alesio, V., Porrini, A. T., Greco, M., & Moro, A. (2025). An investigation of nominal copular sentences in three reading paradigms: Acceptability judgments, self-paced reading, and eye-tracking. *Lingua*, 326, 104019.
- Drummond, A. (2013). IbeX Farm. Available at: <https://farm.pcibex.net/>
- Gračanin-YukseK, M., Lago, S., Şafak, D. F., Demir, O., & Kırkıcı, B. (2017). The interaction of contextual and syntactic information in the processing of Turkish anaphors. *Journal of Psycholinguistic Research*, 46(6), 1397–1425.
- Gračanin-YukseK, M., Lago, S., Şafak, D. F., Demir, O., & Kırkıcı, B. (2020). The interpretation of syntactically unconstrained anaphors in Turkish heritage speakers. *Second Language Research*, 36(4), 475–501.
- Hatzidaki, A., & Santesteban, M. (2024). Emotion effects survive non-standard orthographic representations. *Cognition and Emotion*, 1–8. <https://doi.org/10.1080/02699931.2024.2362381>

- Just, M. A., Carpenter, P. A., & Woolley, J. D. (1982). Paradigms and processes in reading comprehension. *Journal of experimental psychology: General*, 111(2), 228.
- Kim, D., & Kim, H. (2025). Effects of L1 knowledge and task type on bilingual acquisition and processing of English count and mass nouns. *International Journal of Bilingualism*.
<https://doi.org/10.1177/13670069251360171>
- Kim, H., & Hwang, H. (2025). Learning contexts and proficiency matter: L2 real-time sensitivity to conventional and unconventional dative pattern. *Language and Cognition*, 17, e21.
<https://doi.org/10.1017/langcog.2024.65>
- Lee, S. H., & Kaiser, E. (2021). Does hitting the window break it? Investigating effects of discourse-level and verb-level information in guiding object state representations. *Language, Cognition and Neuroscience*, 36(8), 921–940.
<https://doi.org/10.1080/23273798.2021.1896013>
- Liu, J., Fan, L., Zhang, X., & Xu, Z. (2025). Affective and psycholinguistic norms for 1200 two-character Chinese words. *International Journal of Applied Linguistics*.
- Lu, X. (2024). Enhancing the processing advantage: Two psycholinguistic investigations of formulaic expressions in Chinese as a second language. *International Review of Applied Linguistics in Language Teaching*. <https://doi.org/10.1515/iral-2023-0262>
- Moulton, K., Block, T., Gendron, H., Storoshenko, D., Weir, J., Williamson, S., & Han, C.-H. (2022). Bound variable singular *they* is underspecified: The case of *all* vs. *every*. *Frontiers in Psychology*, 13, 880687.
- Ovans, Z., Ayala, M. R., Asmah, R., Hu, A., Montoute, M., Van Horne, A. O., Qi, Z., Morini, G., & Huang, Y. T. (2025). The feasibility of remote visual-world eye-tracking with young children. *Open Mind*, 9, 992–1019. <https://doi.org/10.1162/opmi.a.16>
- Özsoy, O., Çiçek, B., Özal, Z., Gagarina, N., & Sekerina, I. A. (2023). Turkish-German heritage speakers' predictive use of case: webcam-based vs. in-lab eye-tracking. *Frontiers in Psychology*, 14, 1155585.
- Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). *WebGazer: Scalable webcam eye tracking using user interactions*. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16) (pp. 3839–3845). AAAI Press.
<https://dl.acm.org/doi/10.5555/3061053.3061156>
- Park, S.-H., & Lee, E.-K. (2024). Definiteness and context in double-accusative ditransitives in Korean: An experimental approach. *Language Research*, 41(3), 345–366.
- Patterson, A. S., Nicklin, C., & Vitta, J. P. (2025). Methodological recommendations for webcam-based eye tracking: A scoping review. *Research Methods in Applied Linguistics*, 4(3), 100244.

- Plaut-Forckosh, D., & Meir, N. (2025). Insights into linguistic performance: Investigating pragmatic and syntactic violations of Hebrew definiteness through acceptability judgment and self-paced listening and reading tasks. *Applied Psycholinguistics*, 46, e5.
<https://doi.org/10.1017/S0142716425000037>
- Ruda, M., & Huang, N. (2025). Overt pronouns in null subject languages: Experimental investigation of Kashubian, Polish, and Silesian. *Linguistics*.
- Slim, M. S., & Hartsuiker, R. J. (2023). Moving visual world experiments online? A web-based replication of Dijkgraaf, Hartsuiker, and Duyck (2017) using PCIBex and WebGazer.js. *Behavior Research Methods*, 55(7), 3786–3804. <https://doi.org/10.3758/s13428-022-01989-z>
- Uygun, S. (2022). Processing pro-drop features in heritage Turkish. *Frontiers in Psychology*, 13, 988550.
- Yamaguchi, K., & Ohta, S. (2023). Dissociating the processing of empty categories in raising and control sentences: A self-paced reading study in Japanese. *Frontiers in Language Sciences*, 2, 1138749.
- Zehr, J., & Schwarz, F. (2018). PennController for Internet Based Experiments (IBEX). <https://doi.org/10.17605/OSF.IO/MD832>
- Zhang, N., Ren, J., Wang, M., & Jiang, N. (2023). Automatic phonological access among bilinguals with cross-script languages. *Quarterly Journal of Experimental Psychology*, 77(7), 1399–1417.
<https://doi.org/10.1177/17470218231198500>