Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 8(2) (2025) 105-114
DOI: 10.38016/jista.1563512

İnce Ayar ile PyAnnote Kullanılarak Konuşmacı Güncesi Çıkarımı Doğruluğunun Artırılması

Amine Gonca Toprak[1*], Sercan Çepni[2], Şükrü Ozan[2]

[1] Ar-Ge Departmanı, Türk Telekom, Ankara, Türkiye

[2] Ar-Ge Departmanı, AdresGezgini A.Ş., İzmir, Türkiye

goncatoprak@outlook.com, sercancepni@outlook.com, sukruozan@gmail.com

Öz

Konuşma işleme, konuşmacı güncesi çıkarımı, konuşma tanıma ve ses olayı tespiti gibi birçok uygulamada önemli bir rol oynamaktadır. Bu çalışma, firma çağrı merkezi kayıtları üzerinde karşılaştırmalı bir analiz sunmak amacıyla PyAnnote adlı güçlü, açık kaynaklı konuşmacı güncesi çıkarım aracını kullanmaktadır. PyAnnote modelleri, firma çağrı merkezi kayıtlarının etiketlenmesiyle elde edilen veri seti ile ince ayar yapılarak değerlendirilmiş ve baz model performansları ile karşılaştırılmıştır. Model performansları, DER metriği kullanılarak değerlendirilmiş ve sonuç olarak, ince ayar yapılan PyAnnote 3.1 modeli üstün performans göstermiştir. İnce ayar sonrası PyAnnote 3.1 versiyonunun DER puanı %21.9'dan %15.6'ya düşmüştür.

Anahtar kelimeler: konuşmacı güncesi çıkarımı, pyannote, ince ayar

Enhancing Speaker Diarization Accuracy of PyAnnote by Fine-tuning

Abstract

Speech processing plays a crucial role in various applications such as speaker diarization, speech recognition, and sound event detection. This study utilizes PyAnnote, a powerful open-source speaker diarization tool, to conduct a comparative analysis on company call center recordings. PyAnnote models were fine-tuned using a dataset obtained from the annotation of company call center recordings and evaluated against the baseline model performances. Model performances were assessed using the DER metric, and as a result, the fine-tuned PyAnnote 3.1 model demonstrated superior performance. After fine-tuning, the DER score of PyAnnote 3.1 decreased from 21.9% to 15.6%.

Keywords: speaker diarization, pyannote, fine-tuning

1. Giriş (Introduction)

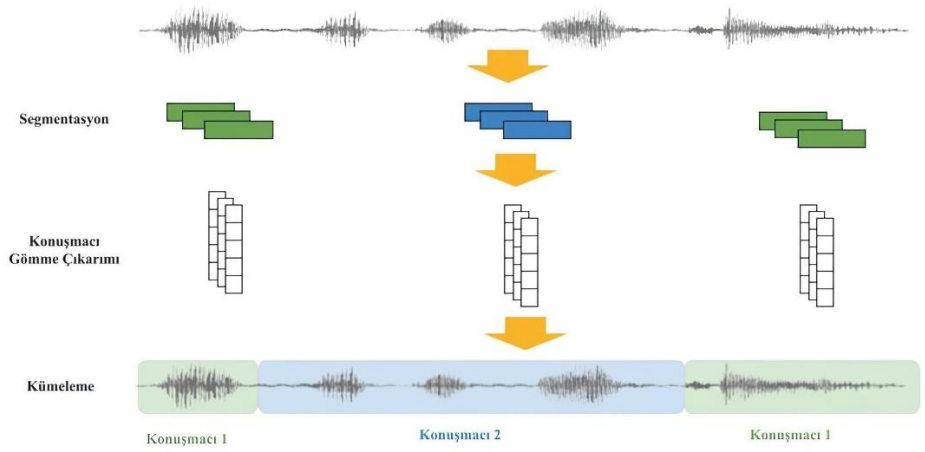
Konuşmacı güncesi çıkarımı, bir ses veya video kaydındaki konuşmacı kimliklerini ve konuşma süreçlerini (yani kimin ne zaman konuştuğunu) belirlemenin bir yöntemi olarak tanımlanabilir. Bu çıkarım, bir konuşma kaydının farklı konuşmacılar tarafından gerçekleştirilen bölümlere ayrılması ve her bir konuşma bölümünün, ilgili bölümde konuşan konuşmacıya atanmasıyla elde edilir. Konuşma işleme sistemlerinin önemli bileşenlerinden biri olan konuşmacı güncesi çıkarımı, ses kayıtlarını her biri farklı konuşmacıyı temsil eden farklı bölümlere ayırmayı amaçlar. Kısaca ifade etmek gerekirse, konuşmacı güncesi çıkarımı, bir ses kaydında hangi konuşmacının ne zaman konuştuğunu belirleme işlemidir. Konuşmacı güncesi çıkarımının temel amacı, aynı konuşmacının konuşma bölümlerini tanımlayarak ve gruplandırarak transkripsiyon, konuşmacı tanıma, konuşmacıya özgü analiz gibi alt uygulamalara olanak sağlamaktır (Zhang vd., 2019).

Konuşmacı güncesi çıkarımı, konuşma aktivitesinin belirlenmesi (voice activity detection), konuşmacı değişikliği algılama (speaker change detection), örtüşen konuşma algılama (overlapped speech detection), özellik çıkarma (feature extraction), kümeleme veya sınıflandırma gibi temel aşamalardan oluşmaktadır. Klasik konuşmacı güncesi çıkarımı sistemleri konuşmacı gömmelerinin (speaker embeddings) kümelenmesine dayanır. Klasik konuşmacı güncesi çıkarımı adımları segmentasyon, konuşmacı

* Sorumlu yazar.
E-posta adresi: goncatoprak@outlook.com

Alındı : 8 Ekim 2024
Revizyon : 6 Şubat 2025
Kabul : 21 Nisan 2025

gömmelerinin çıkarılması, kümeleme ve yeniden segmentasyon olmak üzere dört temel adımda özetlenebilir (Şekil 1). Segmentasyon adımında, ele alınan ses kaydındaki konuşma olmayan kısımları konuşma aktivitesinin belirlenmesi ile kaldırılarak konuşma içeren kısımlar segmentlere ayrılır. Konuşmacı gömmelerinin çıkarılması adımında, ses kaydındaki konuşma içermeyen kısımların kaldırılmasının ardından bir önceki adımda elde edilen segmentlerden konuşmacı gömmeleri çıkarılır. Kümeleme adımında, her ses segmenti için konuşmacı gömmesi elde edildikten sonra bu segmentler farklı gruplara kümelenir. Bu aşamada her bir küme, bir konuşmacı kimliğine denk gelmektedir (Park vd., 2022, Huang vd., 2020, Khoma vd., 2023, Horiguchi vd., 2021).



Şekil 1. Klasik konuşmacı güncesi çıkarımı temel adımları (Typical speaker diarization main steps)

Geleneksel konuşmacı güncesi çıkarımı yaklaşımları genellikle Gauss Karışım Modeli (GMM) veya Saklı Markov Modeli (HMM) gibi tekniklere dayanır ve akustik özellikler olarak spektral analiz veya Mel-Frekans Kepstral Katsayıları (MFCC) kullanır (Ahmad ve Zubair, 2019). Günümüz literatüründe derin öğrenmeye dayalı yaklaşımlar, özellikle Yinelemeli Sinir Ağları (RNN) ve Evrişimli Sinir Ağları (CNN) gibi ardışık modelleme yapıları sayesinde geleneksel yöntemlerin yetersiz kaldığı durumlarda gelişmiş performans sergilemiştir (Sharma vd., 2020).

Literatürde konuşmacı güncesi çıkarımı için farklı dillerde kodlanmış açık kaynaklı yazılımlar bulunmaktadır. Örneğin, S4D (SIDEKIT for diarization) konuşmacı güncesi çıkarımı için Python dilinde kodlanmış açık kaynak kodlu bir kütüphanedir. S4D, akustik özellik çıkarımında MFCC ve GMM gibi teknikleri kullanırken i-vektör tabanlı kümeleme ile konuşmacı güncesi segmentlerini oluşturmaktadır (Broux vd., 2018). KALDI ise konuşmacı tanıma ve ses işleme için C++ dilinde geliştirilmiş olan açık kaynak kodlu bir yazılımdır. KALDI, konuşmacı güncesi çıkarımı için sonuçlar üretebilse de temelde konuşmacı tanıma sistemleri için geliştirildiğinden detaylı bir konuşmacı güncesi çıkarımı sunmamaktadır (Ravanelli vd., 2019). Yine C++ dilinde kodlanmış olan ALIZE uzantısı, konuşmacı güncesi çıkarımı için geliştirilmiştir fakat derin öğrenme yaklaşımları yerine geleneksel teknikler kullandığından güncel literatürün gerisinde kalmaktadır (Bonastre vd., 2005). PyAudioAnalysis ise genel ses işleme için geliştirilmiş olan, MFCC tabanlı konuşmacı güncesi çıkarımı özelliği bulunan bir Python kütüphanesidir. Konuşmacı güncesi çıkarımı için kullanılabilse de geliştirilmeye açık model vb. özellikler sunmamaktadır (Giannakopoulos, 2015). PyAnnote ise Python dilinde konuşmacı güncesi çıkarımı için geliştirilmiş olan açık kaynak kodlu bir Python kütüphanesidir. PyAnnote yapısı, uçtan uca eğitilmiş nöral bloklara dayanmaktadır ve konuşmacı güncesi çıkarımı için geliştirilebilir modeller sunmaktadır (Bredin, 2023).

Bu çalışmada ise çağrı merkezi kayıtlarından konuşmacı güncesi çıkarımı yapabilmek adına iki farklı versiyon için PyAnnote model performansları, firma çağrı merkezi kayıtlarından elde edilen veri seti üzerinde ince ayar yapmadan önce ve sonra olmak üzere karşılaştırılmıştır. Modellerinin performansları Diarization Error Rate (DER) ile ölçülmüştür. Problem özelinde ince ayar ile iyileştirildiği gözlemlenen PyAnnote 3.1 versiyonlu model, %16 DER puanı ile diğer versiyona göre daha başarılı performans göstererek çağrı merkezi kayıtlarında daha iyi bir konuşmacı güncesi çıkarımı yapılmasını sağlamıştır.

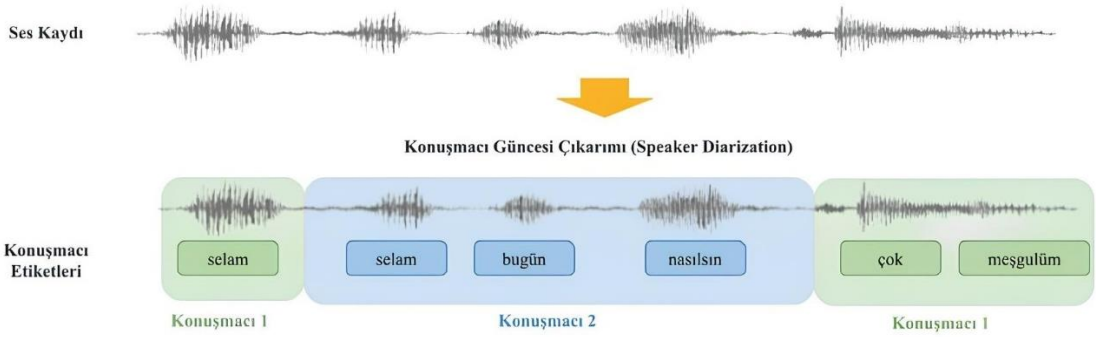
2. Teorik Metod (Theoretical Method)

Bu çalışmada, çağrı merkezi kayıtlarından konuşmacı güncesi çıkarımı yapmak amacıyla

PyAnnote modelinin performansı değerlendirilmiştir. Sunulan iki farklı versiyonu kullanılarak firma çağrı merkezi kayıtlarından elde edilen veri seti üzerinde, ince ayar yapılmadan önceki ve sonraki model performansları karşılaştırılmıştır. Bu amaçla PyAnnote kütüphanesinin sunduğu ses etiketleme aracı ile firma çağrı kayıtlarından 100, 200 ve 400 saat olmak üzere üç farklı eğitim veri seti elde edilmiştir. Test veri seti için ise yaklaşık 7 saatlik firma çağrı merkezi kayıtları manuel olarak etiketlenmiştir. Elde edilen veri seti ile PyAnnote'un sunduğu iki farklı versiyon için baz modellerin performansları, test veri seti üzerinde herhangi bir eğitim gerçekleştirmeden DER metriği ile ölçülmüştür. Sonrasında, elde edilen eğitim veri seti ile her farklı iki versiyon için önceden eğitilmiş baz modeller ince ayar yapılmış ve performansları test verisi üzerinde DER metriği ile ölçülerek karşılaştırılmıştır. Aynı veri seti üzerinde iki farklı model versiyonun performansı, istatiksel olarak Eşleştirilmiş T-Test (Paired T-Test) ile değerlendirilerek DER puanlarındaki değişimin anlamlı olup olmadığı değerlendirilmiştir.

2.1. Konuşmacı Güncesi Çıkarımı (Speaker Diarization)

Konuşmacı güncesi çıkarımı, bir ses kaydının otomatik olarak tanımlama ve ses kaydındaki konuşma içeren her bir bölümün belirli bir konuşmacıya karşılık geldiği farklı konuşma bölümlerine ayırma işlemidir. Daha kısa bir şekilde ifade etmek gerekirse, konuşmacı güncesi çıkarımı, "Kim ne zaman konuştu?" sorusunu yanıtlamayı amaçlar. Temel olarak konuşmacı güncesi çıkarımı sistemleri, ses sinyalini analiz ederek konuşmacı kimliğindeki değişiklikleri tespit eder ve ardından aynı konuşmacıya ait konuşma bölümlerini gruplar (Şekil 2).



Şekil 2. Konuşmacı güncesi çıkarımı illüstrasyonu (Speaker diarization illustration)

Konuşmacı güncesi çıkarımı, ses kayıtlarını otomatik olarak transkribe edebildiğinden konuşma analizi araçlarının önemli bir bileşenidir. Birden fazla konuşmacıyı içeren konuşma analizi gerçekleştirilirken önemli bilgiler sağladığından konuşma içeriğinden anlamlı bilgiler çıkarabilmek adına genellikle Otomatik Konuşma Tanıma (Automatic Speech Recognition - ASR) ve Konuşma Duygusu Tanıma (Speech Emotion Recognition - SER) ile birlikte kullanılmaktadır (Anguera vd., 2012, Tranter ve Reynolds, 2006, Moattar ve Homayounpour, 2012, Garcia-Romero vd., 2017, Xiao vd., 2021).

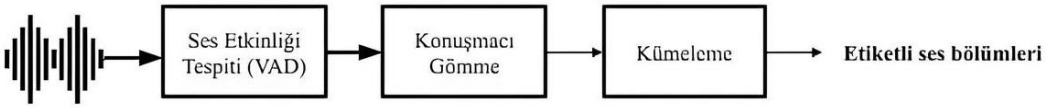
Yapay zekanın yaygınlaşmasından önce, konuşmacı güncesi çıkarımı yöntemleri Dijital Sinyal İşleme teknikleri ile geliştirilirken daha sonra GMM, HMM gibi istatiksel modelleme yöntemlerine dayanmaktaydı (Madikeri ve Bourlard, 2015). Günümüzde ise gelişen teknolojiyle birlikte hemen hemen her alanda yaygınlaşan yapay zeka yöntemleri sayesinde, modern konuşmacı güncesi çıkarımı sistemleri, derin öğrenme ve sinir ağı mimarilerine dayanmaktadır. Dolayısıyla modern konuşmacı güncesi çıkarımı geliştirme yöntemleri temel olarak iki yaklaşım altında ele alınabilir:

- Denetimli öğrenme: Bir sınıflandırma problemi olarak ele alınabilecek sınırlı sayıda konuşmacıyı yani sınıfı tanıyacak şekilde eğitilen konuşmacı güncesi çıkarımı modelidir. Bu nedenle denetimli öğrenme ile geliştirilen konuşmacı güncesi çıkarımı modelinin yalnızca eğitim verisindeki konuşmacılarla çalışması beklenmektedir. Örneğin haftalık takım toplantılarının transkripsiyonu problemi için kullanılabilir. Burada konuşmacı güncesi çıkarımı modeli hep aynı konuşmacılar üzerinden eğitileceğinden daha verimli olabilir.

- Denetimsiz öğrenme: Bir kümeleme problemi olarak ele alınabilecek ilgili konuşmacıların sayısını yani küme sayısını tespit edebilecek ve her ses bölümünü belirli bir kümeye atayabilecek şekilde eğitilen konuşmacı güncesi çıkarımı modelidir. Model, ses özelliklerine dayalı olarak ses bölümlerini konuşmacıya göre kümeleyebilir. Bu nedenle

denetimli öğrenmeye göre daha geniş bir kapsam ve çok yönlülük sunar. Örneğin bir çağrı merkezindeki görüşmelere hangi konuşmacıların dahil olduğunu analiz etmede kullanılabilir. Burada konuşmacı güncesi çıkarımı modeli, konuşmaya dahil olan konuşmacılar hakkında herhangi bir ön bilgiye sahip olmadan kümeleme yöntemiyle konuşmacının müşteri yetkilisi veya müşteri olup olmadığını ayırt edebilir.

Daha geniş bir kullanım alanı olduğundan, güncel literatürde denetimsiz öğrenme ile eğitilen konuşmacı güncesi çıkarımı yaklaşımı daha yaygın olarak kullanılmaktadır. Bu bağlamda tipik bir denetimsiz konuşmacı güncesi çıkarımı boru hattı temel olarak ses etkinliği tespiti (VAD), konuşmacı gömmelernin çıkarılması ve kümeleme veya sınıflandırma olmak üzere çeşitli alt görevleri içermektedir. Tipik bir konuşmacı güncesi çıkarımı boru hattı Şekil 3'te gösterilmiştir.



Şekil 3. Konuşmacı Güncesi Çıkarımı Boru Hattı (Speaker Diarization Pipeline)

Literatürde Ses Etkinliği Tespiti (Voice Activity Detection - VAD) olarak geçen işlem, ses bölümlerinde konuşma olup olmadığının tespit edilmesi olarak tanımlanır (Graf vd., 2015). Bu adımda model, ses bölümlerinde herhangi bir ses etkinliği olup olmadığını ikili sınıflandırma yöntemi ile tespit etmektedir. Konuşmacı güncesi çıkarımı modeline girdi olarak verilen ses kaydındaki konuşma içeren bölümler tanımlandıktan sonra, farklı seslerden anlamlı öznitelikler çıkarabilmek adına konuşma içeren ses bölümlerini kullanan bir konuşmacı gömme çıkarım modeli gerekmektedir. Bu adımdaki amaç, konuşmacıları ayırt edebilmek adına ses bölümlerinden anlamlı bilgiler elde etmektir. Daha sonra elde edilen özniteliklerden benzer olanların kümelenmesi ile ses kaydındaki farklı konuşmacılar tespit edilir. Kümeleme adımında her farklı küme ses kaydındaki benzersiz konuşmacıyı temsil etmektedir.

Tipik bir konuşmacı güncesi çıkarımı sistemi, bu üç temel adım ile konuşmacı güncesi çıkarımı modeline direkt olarak verilen ses kayıtlarında hangi konuşmacının hangi zaman aralıklarında konuştuklarını tespit edebilmektedir.

2.2. Veri Seti (Dataset)

Çalışma kapsamında oluşturulan veri seti standartlarının belirlenmesi için güncel konuşmacı güncesi çıkarımı literatüründe yaygın olarak kullanılan DIHARD 3 (Ryant vd., 2020), AliMeeting (Yu et al., 2022), AMI-SDM (Kraaij et al., 2005), AISHELL-4 (Fu et al., 2021), VoxConverse (Landini et al., 2021) gibi veri setleri incelenerek veri seti oluşturulmasında temel alınmıştır. Güncel literatürde yaygın olarak kullanılan bu veri setleri toplantı, yemek, konferans gibi birden fazla konuşmacının bulunduğu ortamlarda ve kısa duraklama, üst üste binen konuşma, hızlı konuşmacı değişikliği, gürültü gibi gerçekçi doğal konuşma karakteristikleri gözetilerek oluşturulmuştur. Güncel literatürdeki veri setleri, ses kayıtlarına ek olarak konuşmacı güncesi çıkarımı için konuşmacı kimlikleri ve konuşmacı kimliklerinin zaman işaretlerini içeren etiket dosyalarını da sunmaktadır.

Bu çalışma kapsamında oluşturulan veri seti için güncel konuşmacı güncesi çıkarımı literatüründe yaygın olarak kullanılan veri setleri temel alınarak firma çağrı merkezi kayıtları ayıklanmıştır. Bu doğrultuda çalışma kapsamında manuel olarak firma çağrı merkezi ses kayıtlarının etiketlenmesiyle oluşturulacak veri setinin içereceği ses kayıtları için standartlar aşağıdaki gibi belirlenmiştir:

- Aynı cinsiyetten konuşmacıların olduğu ses kayıtları (kadın-kadın, erkek-erkek)
- Farklı cinsiyetten konuşmacıların olduğu ses kayıtları (kadın-erkek)
- Aynı cinsiyetten fakat yakın ses tonuna sahip konuşmacıların olduğu ses kayıtları (kadın-kadın, erkek-erkek)
- Arka planda çok fazla gürültü içermeyen ses kayıtları
- Konuşmaların net bir şekilde anlaşılabildiği ses kayıtları
- Konuşmaların nispeten daha az anlaşılabilir olduğu ses kayıtları
- 5 dakikadan uzun ve farklı uzunluklardaki ses kayıtları

Veri seti oluşturma çalışmaları eğitim ve test veri seti olmak üzere iki farklı aşamada eş zamanlı olarak yürütülmüştür. Eğitim ve test veri setlerinde kullanılan ses kayıtlarının etiketlenmesiyle nihai veri seti elde edilmiştir.

2.2.1. Eğitim Veri Seti (Training Dataset)

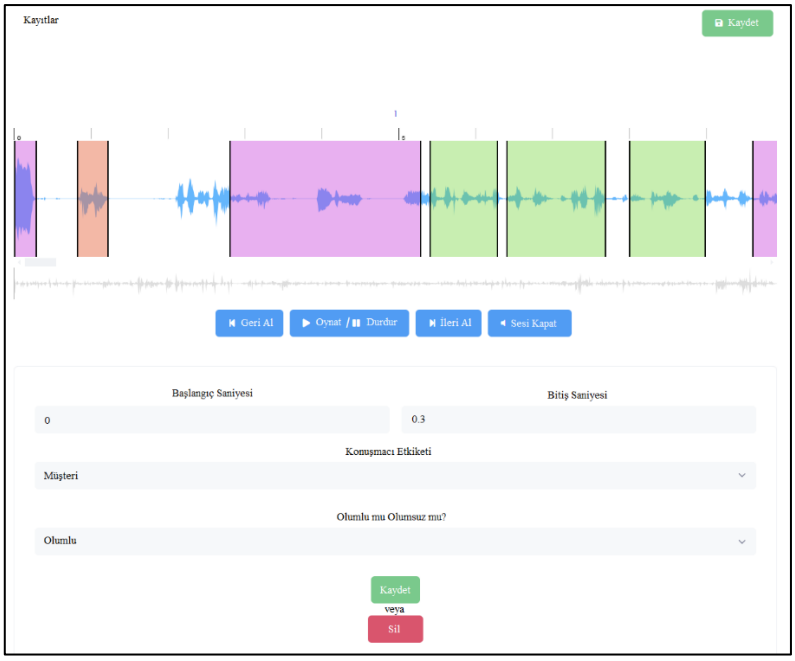
Çalışma kapsamında PyAnnote model eğitimi için kullanılacak eğitim veri setini oluşturmak amacıyla, firma çağrı merkezi kayıtları, PyAnnote ile etiketlenmiş ve manuel olarak kontrol edilerek doğru etiketlenmiş ses

kayıtları ayıklanarak 100, 200 ve 400 saatlik olmak üzere üç farklı eğitim veri seti olacak şekilde elde edilmiştir. Eğitim veri seti oluşturulurken ince ayar eğitiminde veri seti uzunluğunun etkisini de gözlemleyebilmek adına farklı uzunluklarda eğitim veri setlerinin elde edilmesi amaçlanmış ve her farklı eğitim veri seti ile ince ayar eğitimleri gerçekleştirilmiştir. Oluşturulan eğitim veri setleri, PyAnnote'un sunduğu önceden eğitilmiş baz modelinin iki farklı versiyonu için ince ayar yapılmasında kullanılmıştır.

2.2.2. Test Veri Seti (Test dataset)

Baz model ve ince ayar yapılan modellerin performanslarını test etmek amacıyla, firma çağrı merkezi kayıtlarından farklı uzunluklardaki ses kayıtlarının dikkatlice dinlenerek etiketlenmesiyle yaklaşık 7 saatlik bir test veri seti elde edilmiştir. Elde edilen test veri seti etiketlenirken seçilen ses kayıtlarındaki konuşmacıların aynı cinsiyetten olması (kadın-kadın, erkek-erkek), farklı cinsiyetlerden olması (kadın-erkek), aynı cinsiyete ve yakın ses tonlarına sahip olması (sesleri benzeyen kadın-kadın veya erkek-erkek konuşmacılar) gibi kriterlere dikkat edilmiştir.

Çağrı merkezi ses kayıtlarında büyük bir çoğunlukla firma yetkilisi ve müşteri olmak üzere 2 konuşmacı bulunmaktadır. Ses kayıtlarının dinlenerek manuel etiketlenmesi, ses kaydında firma yetkilisinin veya müşterinin konuşmaya başladığı ve konuşmayı bitirdiği bölümlerin ses dalgası üzerinde manuel olarak seçilerek etiketlenmesidir. Çağrı merkezi ses kayıtlarının manuel olarak etiketlenebilmesi için bir ses etiketleme arayüzü geliştirilmiştir. Örneğin ses kaydında müşterinin konuştuğu saniye aralıkları müşteri, firma yetkilisinin konuştuğu saniye aralıkları firma yetkilisi olarak ses etiketleme arayüzünde etiketlenmiştir (Şekil 4). Geliştirilen ses etiketleme arayüzü ile ses kayıtlarının ilgili etiket dosyaları manuel olarak elde edilmiştir.

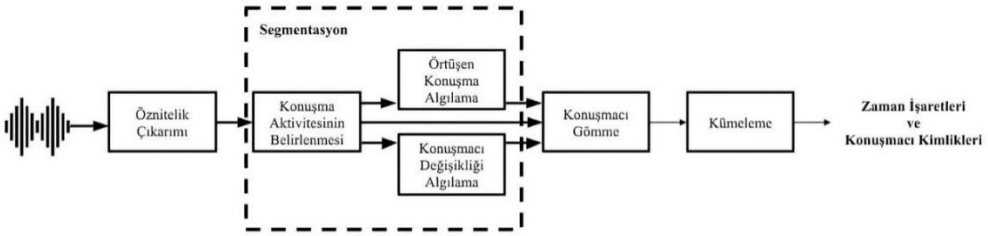


Şekil 4. Ses etiketleme arayüzü (Speech labelling interface)

Geliştirilen ses etiketleme arayüzünde, ses kaydı dinlenerek ilgili konuşma bölümleri seçilir ve ilgili konuşmacı etiketi bu ses bölümüne atanır.

2.3. PyAnnote (PyAnnote)

PyAnnote, doğrudan konuşmacı güncesi çıkarımı (speaker diarization) görevleri için geliştirilmiş açık kaynaklı bir kütüphanedir ve mimarisi, denetimsiz konuşmacı güncesi çıkarımı sisteminin uygulanabileceği eğitilmiş nöral uçtan uca bloklardan oluşmaktadır (Şekil 5). Bu bloklar yeniden ayarlanabilir, diğer bloklarla değiştirilebilir veya yeni bloklar eklenerek genişletilebilir. PyAnnote'un sunduğu bu açık blok yapısı, sistem parametrelerinin iyileştirilmesi için yüksek esneklik ve olanak sağlamaktadır.



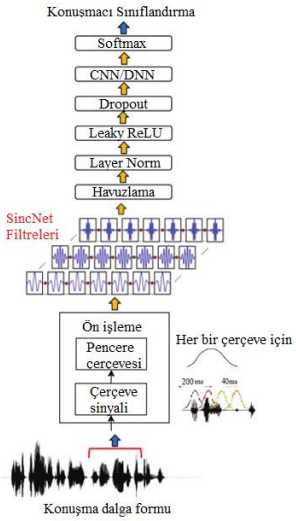
Şekil 5. PyAnnote konuşmacı güncesi çıkarımı boru hattı (PyAnnote speaker diarization pipeline) (Bredin et al., 2020)

Temelde PyAnnote kütüphanesi, sesi dalga formundan doğrudan RTTM formatında konuşmacı etiketlerine dönüştüren bir konuşmacı güncesi çıkarımı boru hattı sunmaktadır. Ek olarak ses işleme ve manipülasyon araçları sunduğundan PyAnnote kütüphanesi, güncel ses işleme ve konuşmacı güncesi çıkarımı literatüründe sıklıkla kullanılmaktadır (Bredin et al., 2020). PyAnnote'un önceden eğitilmiş model versiyonları, AISHELL-4, AliMeeting ve AMI-SDM gibi farklı senaryolar içeren veri setleri üzerinde eğitilmiş olup, bu durum modelin çeşitli senaryolarda başarıyla kullanılabildiğini göstermektedir (Bredin, 2023).

PyAnnote mimarisinde, her bloğun genel çalışma prensibi, girdinin öznitelik vektörleri ve beklenen çıktının ilgili etiket dizisi olduğu bir dizi etiketleme (sequence labeling) görevi olarak tanımlanabilir. PyAnnote kütüphanesi, bir öznitelik dizisini ilgili etiket dizisine eşleyen önceden eğitilmiş PyTorch modellerinden oluşan bir mimariye dayanmaktadır.

2.3.1. Öznitelik Çıkarma Bloğu (Feature Extracrion Block)

PyAnnote, özellik çıkarma modülü olarak SincNet adlı evrişimli sinir ağı mimarisini kullanmaktadır. SincNet, konuşma işleme görevlerinde kullanılan bir evrişimli sinir ağı (CNN) mimarisidir. Bu özel mimari konuşma sinyallerinden özellik çıkarmak amacıyla tasarlanmıştır (Ravanelli ve Bengio, 2018). Şekil 6, SincNet mimarisini temsil etmektedir. SincNet'in ana amacı, konuşma sinyallerinden anlamlı özellikler çıkarmaktır. Bu özellikler daha sonra konuşmacı tanıma, konuşma sınıflandırma veya diğer konuşma işleme görevlerinde kullanılmaktadır. SincNet, sinc fonksiyonlarını kullanarak düşük frekansta özellik çıkarmak üzere tasarlanmıştır. Sinc fonksiyonları, belirli frekanstaki bileşenleri çıkarmak veya belirli bir frekansta bilgiyi vurgulamak için kullanılır. Bu sayede, özellikle konuşma sinyallerinin temel frekans bileşenlerini çıkarmak için etkili bir şekilde kullanılır.



Şekil 6. SincNet mimarisi (SincNet architecture) (Ravanelli ve Bengio, 2018)

2.3.2. Segmentasyon Bloğu (Segmentation Block)

Segmentasyon bloğundaki tüm algılama modülleri (Tablo 1), örtüşen kayan pencereler kullanılarak kısa sabit uzunluklu alt diziler üzerinde gerçekleştirilir. Bu, birden fazla tahmin puanıyla sonuçlanır ve bunların ortalaması bir final tahmin puanına dönüştürülür.

Tablo 1. Segmentasyon bloğundaki algılama modüllerinin işlevleri (Functions of the detection modules in the segmentation block)

Modül	İşlevi
Ses Etkinliği Tespiti	Ses kaydında insan sesinin bulunduğu zaman aralıklarını tespit etmek
Örtüşen Konuşma Algılama	Ses kaydında iki veya daha fazla konuşmacının aynı anda konuştuğu zaman aralıklarını tespit etmek
Konuşmacı Değişikliği Algılama	Ses kaydında bir konuşmacının konuşmasının bittiği ve başka birinin konuşması başladığı anları tespit etmek

Her üç modül için de başlatılabilen ayarlanabilir bir eşik vardır. Her t zaman adımında, tahmin puanı önceden tanımlanmış eşikle karşılaştırılır ve eşik aşıldığında 1, aksi halde 0 puanı elde edilir. 1 puanı, sesin, konuşmacı değişikliğinin veya örtüşen konuşmanın algılandığını belirtir (Bredin ve Laurent, 2021). PyAnnote mimarisi, Segmentasyon bloğunun yeniden eğitilmesi ile ince ayar yapılabilmesini mümkün kılmaktadır.

2.3.3. Konuşmacı Gömme ve Kümeleme Blokları (Speaker Embedding and Clustering Blocks)

Konuşmacı gömme modülü, belirli bir boyuta sahip bir vektör oluşturmaya yönelik bir modüdür ve bunun sayısal değerleri, fizyolojik ses parametreleri temelinde türetilen konuşmacıya özgü özellikleri temsil eder. Bu blokta, x-vektörleri kullanılarak konuşmacı gömmeleri (speaker embedding) oluşturulur. Bunlar, kosinüs mesafe metriklerine (cosine distance metrics), merkez bağlantıya (central linkage) ve hiyerarşik birleştirici (hierarchical agglomerative clustering) kümeleme dayanan kümeleme aşaması için girdi olarak kullanılır. Böylece bir kişiye ait cümlelere karşılık gelen vektörler arasındaki mesafe, diğer konuşmacıların söylediği cümlelerden oluşturulan vektörlere olan mesafeden daha küçüktür. Kümeleme bloğunda ise konuşmacılara karşılık gelen bölümlerin konuşmacı gömmelerini kullanılarak gruplandırılır. Böylelikle ses kayıtlarındaki konuşmacılara ait zaman damgaları ve konuşmacı kimlikleri elde edilir (Mul, 2023).

Sonunda, her yapı bloğu bağımsız olarak eğitildikten sonra optimize edilmiş hiperparametrelere sahip bir boru hattında (pipeline) birleştirilir. Sesli konuşma alanlarını içeren farklı veri kümeleri üzerinde eğitilir, ayarlanır ve test edilir. Bu, özellikle benzer ayarlardan gelen ses için boru hattını sağlam ve genelleştirilebilir hale getirir. Boru hattı kullanıma hazırdır ancak probleme özel veriler kullanılarak da ince ayar yapılabilir (Yin et al., 2018). Konuşmacı güncesi çıkarımı zorluklarına sunduğu kullanımı kolay ve ince ayar yapılabilir bir boru hattı sunduğu için PyAnnote kütüphanesi güncel literatürde yaygın olarak kullanılmaktadır.

Literatürde Kaldi, DeepSpeaker, SD4 gibi konuşmacı güncesi çıkarımı için farklı yaklaşımlar bulunsa da, bu yaklaşımların sınırlamalarından dolayı (Tablo 2) çalışma kapsamında PyAnnote mimarisi kullanılmıştır.

Tablo 2. Konuşmacı Güncesi Çıkarımı Sistemleri Karşılaştırma Tablosu (Comparison of Speaker Diarization Systems)

Sistem	Kullanım Alanı	Avantajlar	Sınırlandırmalar
Kaldi	Ses tanıma ve konuşmacı ayrıştırma	Daha geniş topluluk desteği, esnek yapılandırma	Eğitim süreci karmaşık ve daha fazla manuel ayar gerektiriyor
DeepSpeaker	Konuşmacı doğrulama	Hafif model yapısı, düşük işlem maliyeti	Büyük veri setleri gerektiriyor, sınırlı özellikler
S4D	Konuşmacı güncesi çıkarımı ve tanıma	Özelleştirilebilir yapı, derin öğrenme tabanlı yaklaşımlar	Kaynak tüketimi yüksek, optimize edilmesi zor
PyAnnote	Genel konuşmacı güncesi çıkarımı	Açık kaynak, güçlü önceden eğitilmiş modeller, ince ayar yapılabilir	İnce ayar süreci veri setine özel optimizasyon gerektiriyor

2.4. Model Performans Ölçümü (Model Performance Evaluation)

Güncel literatürde konuşmacı güncesi çıkarımı performansını ölçümlemek adına Diarization Error Rate (DER) puanı kullanılmaktadır. Bu çalışmada da konuşmacı güncesi çıkarımı performansını ölçümleyebilmek adına DER puanı kullanılmıştır.

2.4.1. Konuşmacı Güncesi Çıkarımı Hata Oranı (Diarization Error Rate - DER)

DER, konuşmacı güncesi çıkarımındaki hatayı süre oranı cinsinden hesaplayan bir metriktir (Anguera vd., 2012). DER aşağıdaki gibi hesaplanır;

$$\text{DER} = \frac{FA+MISS+ER}{\text{TOTAL}} \qquad (1)$$

DER metriği konuşmacı güncesi çıkarımındaki hatayı süre oranı cinsinden hesaplar. DER metriği hesaplanırken referans etiketlerle konuşmacı güncesi çıkarımı çıktısı karşılaştırılarak Tablo 3'teki üç hata teriminin toplam konuşmacı süresine yani tüm referans konuşmacı bölüm sürelerinin toplamına (TOTAL) oranlanması ile hesaplanır.

Tablo 3. DER metriği hesaplamasında kullanılan hata metrikleri (Error metrics used in DER metric calculation)

Hata Metriği	Tanım
Yanlış Alarm (False Alarm - FA)	Konuşmacının olmadığı yerde tahmin edilen konuşma bölümü (VAD modelindeki yanlış pozitif)
Yanlış Tespit (Missed Detection - MISS)	Konuşmacının olduğu yerde konuşma algılanmaması (VAD modelindeki yanlış negatif)
Hata (Error - ER)	Konuşma bölümünün yanlış kümeye atanması (Kümeleme modelindeki hata)

DER değerinin mümkün olduğunca küçük olması modelin konuşmacı güncesi çıkarımı performansının başarısını yani modelin kayıtlardaki konuşmacıları ne kadar iyi ayırt edebildiğini yorumlamamızı sağlar.

2.4.2. Eşleştirilmiş T-Testi (Paired T-Test)

Eşleştirilmiş T-Testi, aynı örneklem üzerinde yapılan iki farklı ölçüm arasındaki farkın istatiksel olarak anlamlı olup olmadığını analiz etmek için kullanılan parametrik bir testtir. Bir deney öncesi ve sonrası ölçülen değerler veya aynı veri noktaları üzerinde farklı koşullarda elde edilen sonuçlar arasındaki farkları analiz etmek için kullanılır. Sıfır hipotezi (H_0), iki ölçüm arasındaki farkın 0 olduğunu, alternatif hipotez ise (H_A) bu farkın anlamlı olup olmadığını iddia eder. T-istatistiği, iki ölçüm arasındaki farkların ortalamasının standart hata ile bölünmesiyle hesaplanır ve p-değeri aracılığıyla istatistiksel anlamlılık değerlendirilir. $p < 0.05$ olması durumunda, iki ölçüm arasındaki farkın istatistiksel olarak anlamlı olduğu kabul edilir.

3. Sonuçlar ve Tartışmalar (Results and Discussions)

Bu çalışmada, PyAnnote mimarisinin firma çağrı merkezi kayıtları üzerindeki performansına odaklanılmıştır. Çalışma kapsamında, firma çağrı merkezi kayıtlarında müşteri ve müşteri yetkilerinin bir konuşma kaydında hangi sürelerle konuştuğunu analiz edebilmek adına PyAnnote'un sunduğu önceden eğitilmiş konuşmacı güncesi çıkarımı modelleri, firma çağrı merkezi kayıtları ile ince ayar yapılması hedeflenmiştir. Bu amaç doğrultusunda firma çağrı merkezi kayıtlarının PyAnnote ile etiketlenmesi ile 100, 200 ve 400 saatlik olmak üzere üç farklı eğitim veri seti, belirlenen kriterlere göre seçilen ses kayıtlarının manuel olarak etiketlenmesi ile ise yaklaşık 7 saatlik test veri seti elde edilmiştir. İnce ayarın model performansına etkisini ölçümleyebilmek adına öncelikle PyAnnote'un sunduğu iki farklı önceden eğitilmiş baz model versiyonları, elde edilen test veri setinde ölçülmüştür. Daha sonra, elde edilen eğitim veri seti ile her farklı versiyon baştan eğitilerek ince ayar yapılmıştır. Model performansları DER puanı kullanılarak değerlendirilmiştir (Tablo 4). DER puanı hesaplanırken kullanılan FA, MISS ve ER oranları, baz model ve ince ayar süreci boyunca gözlemlenmiş ve sonuçların değerlendirilmesinde kullanılmıştır. Yapılan analizde, ince ayarlı modelin tüm veri seti boyutlarında FA, MISS ve ER değerlerini düşürerek bu iyileşmelerin, ince ayarın model performansına olumlu katkı sağladığını göstermiştir.

Önceden eğitilmiş PyAnnote modellerini ince ayar yapabilmek için elde edilen 100, 200 ve 400 saatlik gerçek çağrı merkezi kayıtları ve etiketleri kullanılmıştır. PyAnnote'un sunmuş olduğu 2.1 ve 3.1 versiyonlu iki farklı baz modelin öncelikle manuel etiketleme ile yine gerçek çağrı merkezi kayıtlarından elde edilen yaklaşık 7 saatlik test verisi üzerinde performansları ölçülmüştür. Model performansını ölçebilmek adına DER puanı kullanılmıştır.

Tablo 4. PyAnnote baz model ve ince ayar performans karşılaştırması (Performance comparison of pyannote base models and fine-tuned models)

Eğitim Veri Seti (h)	PyAnnote Versiyonu	Baz Model FA (%)	Baz Model MISS (%)	Baz Model ER (%)	Baz Model DER (%)	İnce Ayar FA (%)	İnce Ayar MISS (%)	İnce Ayar ER (%)	İnce Ayar DER (%)
100	2.1	10.76	9.41	6.72	26.9	8.76	7.86	5.48	21.9
	3.1	8.76	7.66	5.48	21.9	7.04	6.16	4.4	17.6
200	2.1	10.76	9.41	6.72	26.9	8.04	7.04	5.03	20.1
	3.1	**8.76**	**7.66**	**5.48**	**21.9**	**6.24**	**5.46**	**3.9**	**15.6**
400	2.1	10.76	9.41	6.72	26.9	8.84	7.71	5.52	22.1
	3.1	8.76	7.66	5.48	21.9	8.76	7.66	5.48	21.9

İnce ayar yapılmadan önce PyAnnote 2.1 versiyonu ile %26.9 DER puanı ve 3.1 versiyonu ile %21.9 DER puanı elde edilmiştir. Ardından, önceden eğitilmiş her iki model, 100, 200 ve 400 saatlik olmak üzere elde edilen eğitim veri setleri ile 200 epokta eğitilerek ince ayar yapılmıştır. PyAnnote mimarisi, blok yapılarından oluşmakta olup, ince ayar sürecinde yalnızca segmentasyon bloğunun özelleştirilmesine olanak

tanımaktadır. Bu nedenle, çalışma kapsamında elde edilen veri setleri kullanılarak, mimarinin segmentasyon bloğu eğitilmiştir. Eğitim süreci, PyTorch Lightning kullanılarak gerçekleştirilmiş ve optimizasyon algoritması olarak, PyTorch Lightning'in varsayılan olarak kullandığı Adam optimizer tercih edilmiştir (Tablo 5).

Tablo 5. Model ince ayar süreci (Model fine-tuning process)

Kriter	Açıklama
İnce Ayar Yapılan Blok	Segmentasyon Bloğu
Optimizasyon Algoritması	Adam Optimizer
Epok Sayısı	200

İnce ayar yapılan modellerin performansı, elde edilen yaklaşık 7 saatlik test verisi üzerinde DER puanı ile ölçülmüştür. İnce ayar yapılan modellerin, elde edilen üç farklı uzunluktaki eğitim veri setleri ile eğitilip farklı uzunluktaki her eğitim veri seti bazında performansları incelendiğinde modellerin ince ayar sonucunda performanslarının arttığı gözlemlenmiştir. 100 saatlik eğitim veri seti ile ince ayar yapılan modeller arasından 3.1 versiyonu DER puanı %21.9'dan %17.6'ya düşerek 2.1 versiyonuna göre daha iyi performans göstermiştir. 200 saatlik eğitim veri seti ile ince ayar yapılan modeller arasından 3.1 versiyonu DER puanı %21.9'dan %16'ya düşerek 2.1 versiyonuna göre daha iyi performans göstermiştir. 400 saatlik eğitim veri seti ile ince ayar yapılan modeller arasından 3.1 versiyonu DER puanı %21.9'dan %18.6'ya düşerek 2.1 versiyonuna göre daha iyi performans göstermiştir. DER puanında gözlemlenen iyileşmelerin istatistiksel olarak ne kadar anlamlı olduğunu ölçmek amacıyla Eşleşmeli T-Testi uygulanmıştır. Bu istatistiksel yöntemle baz model ve ince ayar yapılmış modellerin DER puanları arasındaki değişimler analiz edilmiştir. Eşleşmeli T-Testi sonucunda T-istatistiği değeri 4.6 ve p-değeri 0.0058 olarak hesaplanmıştır. P-değeri, istatistiksel anlamlılık eşiği olarak kabul edilen 0.05'ten küçük olduğu için ince ayar ile gerçekleştirilen DER puanlarındaki iyileşmenin istatistiksel olarak anlamlı olduğu belirlenmiştir. Elde edilen sonuçlar doğrultusunda, modellerin ince ayar yapılmasıyla DER puanının düştüğü yani konuşmacı güncesi çıkarımındaki başarım oranlarının arttığı gözlemlenmiştir. PyAnnote'un sunduğu farklı iki versiyon için baz model ve ince ayar yapılmış model performansları karşılaştırıldığında, firma çağrı merkezi kayıtları üzerinde PyAnnote 3.1 versiyonunun 200 saatlik eğitim verisi ile ince ayar yapılmış hali, %15.6 DER puanı ile test veri seti üzerinde en iyi performansı göstermiştir.

Gerçekleştirilen ince ayar deneyleri doğrultusunda PyAnnote baz modellerinin konuşmacı güncesi çıkarımı performansı, gerçek çağrı merkezi kayıtları ile arttırılabildiği gözlemlenmiştir. Çalışmada 100, 200 ve 400 saatlik veri setleri kullanılarak eğitim gerçekleştirilmiş olup, bu veri büyüklükleri literatürdeki benzer çalışmalar dikkate alınarak belirlenmiştir. Ancak, 400 saatlik veri seti ile model başarımında gözlemlenen düşüşün birkaç olası sebebi bulunmaktadır. Öncelikle, modelin kapasitesinin veri artışıyla nasıl değiştiği değerlendirildiğinde, belirli bir noktadan sonra aşırı öğrenme veya eksik öğrenme gibi sorunların ortaya çıkabileceği göz önünde bulundurulmalıdır. Ayrıca, 400 saatlik veri setinin daha fazla çeşitlilik içermesi nedeniyle modelin öğrenme sürecini zorlaştırması, gürültülü veya model için zorlayıcı örneklerin artış göstermesi gibi faktörler de başarım düşüşüne katkıda bulunmuş olabilir. En iyi sonuçların 200 saatlik eğitim veri setinde elde edilmesi, bahsedilen durumlara bağlı olabilir.

Bununla birlikte, mevcut model yapılandırmalarının büyük veri setleriyle beklenenden farklı performans göstermesi, modelin büyük ölçekli veriyle daha etkili öğrenmesini sağlamak için ek iyileştirmeler yapılması gerektiğini göstermektedir. Bu doğrultuda, PyAnnote baz modellerinin ince ayar yapılarak konuşmacı güncesi performansının artırılması umut vadettiğinden, gelecekteki çalışmalarda modelin büyük veri setlerine daha iyi uyum sağlaması için hiperparametre optimizasyonu, veri temizleme ve etiketleme süreçlerinin iyileştirilmesi gibi çeşitli stratejiler uygulanacaktır. Bu bağlamda, daha büyük veri setleri oluşturulurken veri çeşitliliği ve etiket kalitesi gibi faktörler daha fazla gözetilecek ve modelin büyük veri setleriyle daha başarılı genelleme yapabilmesi hedeflenecektir. Ek olarak farklı çağrı merkezlerinden elde edilecek veri setlerinin de kullanılmasıyla birlikte daha genelleştirilebilir sonuçlar elde edilebilir. Ayrıca firma bünyesinde geliştirilmiş olan makine öğrenimi ve doğal dil işleme teknikleriyle konuşma içeriğinin derinlemesine analizi ve duygu analizi gerçekleştiren modüllere, ince ayar yapılan PyAnnote modelinin entegre edilmesiyle müşteri memnuniyetini artırarak çağrı merkezi performansının artırılması amaçlanmaktadır.

Teşekkür (Acknowledgment)

Bu çalışma TÜBİTAK TEYDEB 1501 kapsamında desteklenen 3221247 numaralı "Ses Kayıtlarındaki Konuşmacı Sayısının Belirlenmesini ve Konuşmacı Seslerinin Ayrıştırılmasını Sağlayan Derin Yapay Sinir Ağı Mimarisi Tabanlı Sistemin Geliştirilmesi" isimli proje kapsamında gerçekleştirilmiştir.

Kaynaklar (References)

Ahmad, R., Zubair, S., 2019. Unsupervised deep feature embeddings for speaker diarization. Turkish Journal of Electrical Engineering and Computer Sciences, 27(4), 3138-3149. DOI:10.3906/elk-1901-125

Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., Vinyals, O., 2012. Speaker diarization: A review of recent research. IEEE Transactions on Audio, Speech, and Language Processing, 20(2), 356-370. HAL Id: hal-00733397

Bonastre, J.-F., Wils, F., Meignier, S., 2005. ALIZE, a free toolkit for speaker recognition. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), 1, I/737-I/740. DOI:10.1109/ICASSP.2005.1415219

Bredin, H., 2023. pyannote.audio 2.1 speaker diarization pipeline: Principle, benchmark, and recipe. 24th Interspeech Conference (INTERSPEECH 2023), 1983-1987. HAL Id: hal-04247212

Bredin, H., Laurent, A., 2021. End-to-end speaker segmentation for overlap-aware resegmentation. arXiv:2104.04045

Bredin, H., Yin, R., Coria, J.M., Gelly, G., Korshunov, P., Lavechin, M., Gill, M.P., 2020. Pyannote.audio: Neural building blocks for speaker diarization. ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 7124-7128. arXiv:1911.01255v1

Broux, P.-A., Desnous, F., Larcher, A., Petitrenaud, S., Carrive, J., Meignier, S., 2018. S4D: Speaker Diarization Toolkit in Python. Interspeech 2018, 1368-1372. HAL Id: hal-02280162

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Wellner, P., 2005. The AMI meeting corpus: A pre-announcement. International Workshop on Machine Learning for Multimodal Interaction, 28-39. DOI:10.1007/11677482_3

Fu, Y., Cheng, L., Lv, S., Jv, Y., Kong, Y., Chen, Z., Chen, J., 2021. Aishell-4: An open source dataset for speech enhancement, separation, recognition, and speaker diarization in a conference scenario. arXiv:2104.03603

Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., McCree, A., 2017. Speaker diarization using deep neural network embeddings. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4930-4934. DOI:10.1109/ICASSP.2017.7953094

Giannakopoulos, T., 2015. pyaudioanalysis: An open-source python library for audio signal analysis. PloS One, 10(12), e0144610. DOI:10.1371/journal.pone.0144610

Graf, S., Herbig, T., Buck, M., Schmidt, G., 2015. Features for voice activity detection: A comparative analysis. EURASIP Journal on Advances in Signal Processing, 2015(1), 1-15. DOI:10.1186/s13634-015-0277-z

Horiguchi, S., Garcia, P., Fujita, Y., Watanabe, S., Nagamatsu, K., 2021. End-to-end speaker diarization as post-processing. ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 7188-7192. arXiv:2012.10055v2

Huang, Z., Watanabe, S., Fujita, Y., García, P., Shao, Y., Povey, D., Khudanpur, S., 2020. Speaker diarization with region proposal network. ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6514-6518. arXiv:2002.06220v1

Khoma, V., Khoma, Y., Brydinskyi, V., Konovalov, A., 2023. Development of supervised speaker diarization system based on the PyAnnote Audio Processing Library. Sensors, 23(4), 2082. DOI:10.3390/s23042082

Landini, F., Glembek, O., Matějka, P., Rohdin, J., Burget, L., Diez, M., Silnova, A., 2021. Analysis of the BUT diarization system for VoxConverse challenge. ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5819-5823. arXiv:2010.11718v2

Madikeri, S., Bourlard, H., 2015. KL-HMM based speaker diarization system for meetings. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4435-4439. DOI:10.1109/ICASSP.2015.7178809

Moattar, M.H., Homayounpour, M.M., 2012. A review on speaker diarization systems and approaches. Speech Communication, 54(10), 1065-1103. DOI:10.1016/j.specom.2012.05.002

Mul, A., 2023. Enhancing Dutch audio transcription through integration of speaker diarization into the automatic speech recognition model Whisper. Master's thesis, Utrecht University, Applied Data Science.

Park, T.J., Kanda, N., Dimitriadis, D., Han, K.J., Watanabe, S., Narayanan, S., 2022. A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72(1), 101317. arXiv:2101.09624v4

Ravanelli, M., Bengio, Y., 2018. Speaker recognition from raw waveform with SincNet. IEEE Spoken Language Technology Workshop (SLT), 1021-1028. DOI:10.1109/SLT.2018.8639585

Ravanelli, M., Parcollet, T., Bengio, Y., 2019. The pytorch-kaldi speech recognition toolkit. ICASSP 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6465-6469. arXiv:1811.07453v2

Ryant, N., Singh, P., Krishnamohan, V., Varma, R., Church, K., Cieri, C., Liberman, M., 2020. The third DIHARD diarization challenge. arXiv:2012.01477

Sharma, V., Zhang, Z., Neubert, Z., Dyreson, C., 2020. Speaker diarization: Using recurrent neural networks. arXiv:2006.05596

Tranter, S.E., Reynolds, D.A., 2006. An overview of automatic speaker diarization systems. IEEE Transactions on Audio, Speech, and Language Processing, 14(5), 1557-1565. DOI:10.1109/TASL.2006.878256

Xiao, X., Kanda, N., Chen, Z., Zhou, T., Yoshioka, T., Chen, S., Gong, Y., 2021. Microsoft speaker diarization system for the VoxCeleb speaker recognition challenge 2020. ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5824-5828. arXiv:2010.11458v2

Yin, R., Bredin, H., Barras, C., 2018. Neural speech turn segmentation and affinity propagation for speaker diarization. Interspeech 2018, 1393-1397. DOI:10.21437/Interspeech.2018-1750

Yu, F., Zhang, S., Fu, Y., Xie, L., Zheng, S., Du, Z., Bu, H., 2022. M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge. ICASSP 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6167-6171. arXiv:2110.07393v3

Zhang, A., Wang, Q., Zhu, Z., Paisley, J., Wang, C., 2019. Fully supervised speaker diarization. ICASSP 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6301-6305. arXiv:1810.04719