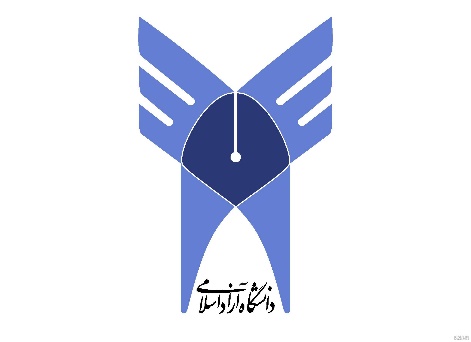
****

**دانشگاه آزاد اسلامی**

**واحد علوم و تحقیقات**

**دانشکده برق و کامپیوتر و مکانیک**

**گزارش کار پروژه نهایی درس گفتار پردازی رقمی**

**موضوع پیاده سازی ASR – WhisperX by (OpenAi Inc)**

**گردآورنده: صبا حصارکی**

**استاد: دکتر کوچاری**

**بهمن 1401**

**Open Ai Whisper**

OpenAI’s Whisper یک مدل جدید پیشرفته (SotA) در گفتار به متن است. این می تواند تقریباً بدون نقص گفتار را در ده ها زبان رونویسی کند و حتی کیفیت صدای ضعیف یا نویز بیش از حد پس زمینه را مدیریت کند.

دامنه کلمات گفتاری همیشه تا حدودی برای موارد استفاده ML دور از دسترس بوده است. Whisper آن را برای موارد استفاده با محوریت گفتار تغییر می دهد. در ابتدا Whisper را در کنار transformer و جستجوی برداری با ایجاد یک جستجوی جدید و بهبودیافته YouTube نشان می دهیم و سپس مدلی به نامwhisperx که تعمیمی از whisper است را معرفی می کنیم که چندین زبان مختلف را پوشش می دهد .

جستجو در یوتیوب خوب است، اما محدودیت‌های خود را دارد، مخصوصاً در مورد پاسخ دادن به سؤالات. با تریلیون ها ساعت محتوا، تقریباً برای هر سؤالی باید پاسخی وجود داشته باشد. با این حال، اگر سؤال خاصی داشته باشیم، به جای یک پاسخ مختصر، ویدیوهای بسیار طولانی زیادی دریافت می کنیم که باید از طریق آنها تماشا کنیم.

اگر تنها چیزی که می خواهیم یک توضیح کوتاه 20 ثانیه ای باشد چه؟ جستجوی فعلی YouTube هیچ راه حلی برای این ندارد. شاید دلیل خوبی برای تشویق کاربران به تماشای هر چه بیشتر یک ویدیو وجود داشته باشد

Whisper راه حل این مشکل و بسیاری موارد دیگر است که شامل کلام گفتاری است.

داده های ویدیویی

اولین قدم این است که داده‌های ویدیوی YouTube خود را دانلود کنیم و صدای attach به هر ویدیو را extract کنید. با pytube))

از pytube، یک video ID (که در URL دیده می‌شود) استفاده می‌کنیم. برای این کار، ابتدا باید صدای ویدیوهایمان را به متن رونویسی کنیم. یوتیوب به‌طور خودکار برای هر ویدیو زیرنویس می‌کند، و زیرنویس‌ها مشکلی ندارند – اما OpenAI فقط چیزی به نام «Whisper» را opensource کرده است.

Whisper بهتر است به عنوان GPT-3 یا DALL-E 2 گفتار به متن توصیف شود. این open source است و می تواند صدا را در زمان واقعی یا سریعتر با عملکرد بالایی transcribe کند.

**قطعه کد**

ابتدا از ریپازیتوری مربوطه کد whisper را نصب می کنیم

|  |
| --- |
| !pip install git+https://github.com/openai/whisper.git -q |

مرحله ی بعد pytube را برای استخراج ویدیو از یوتیوب نصب می کنیم

|  |
| --- |
| !pip install pytube -q |

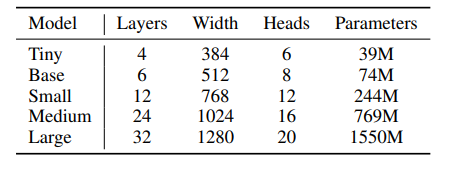
|  |
| --- |
| from pytube import YouTube yt = YouTube('https://www.youtube.com/watch?v=ft-nV2MJPfY') print(yt.title) |

در مرحله ی بعد یک تابع برای دریافت یک ویدیو با فرمت هایی مثل mp4,mov,… نوشته شده و آن ها را تحت پردازش فشرده سازی فرمت ffmpeg به فایل صوتی mp3 تبدیل می کند

|  |
| --- |
| import os  import subprocess   def video2mp3(video\_file, output\_ext="wav"):     filename, ext = os.path.splitext(video\_file)     subprocess.call(["ffmpeg", "-y", "-i", video\_file, f"{filename}.{output\_ext}"],stdout=subprocess.DEVNULL,stderr=subprocess.STDOUT)     return f"{filename}.{output\_ext}" |

حالا یک ویدیو که مکالمه ای با زبان آلمانی به مدت پنج دقیقه هست را به عنوان ورودی داده و تابع بالا را به آن اعمال می کنیم

|  |
| --- |
| input\_video = '/content/drive/MyDrive/1e57321a8807173f30318f02d025647627751531-240p.mp4' audio\_file = video2mp3(input\_video) |

در مدل whisper پنج نوع مدل tiny , base , small , medium ,large داریم که مقادیر پارامتر ها در زیر نشان داده شده است . ابتدا مدل را با سایز mediumلود می کنیم و سپس transcribe ویدیو زبان مبدا (در اینجا آلمانی )را چاپ می کنیم

|  |
| --- |
| import whisper model = whisper.load\_model('medium') model.transcribe(audio\_file,fp16=False)['text'] |

**خروجی**

|  |
| --- |
| Musik Herzlich willkommen zurück bei Easy German. Wir sind heute in New York. Wie ihr wisst, machen wir eine vierwöchige USA-Reise. Und das ist unsere erste Station. Bei uns sind Easy. Hi Easy! Hallo! Ben? Hallo! Unser Übersetzer und Mara. Hi Mara! Hi! Woher kommst du? Ich komme aus Pennsylvania, aber ich studiere an der Uni in New York. Wie kommst du, dass du so gut Deutsch sprichst? Ich lerne Deutsch seit vier Jahren. Sehr gut. Und wir vier gehen heute mal durch New York und suchen dabei Deutsche. Wir wollen Interviews machen, erfahren, was die Menschen hier machen. Mal sehen, wen wir auf dem Weg treffen. Los geht's! Wir haben jetzt gerade vier Deutsche getroffen am Times Square. Sie haben sich dadurch verraten, dass sie Deutsch gesprochen haben. Wo kommt ihr denn her? Nähe Bielefeld. Und wie lange seid ihr schon in den USA? Gestern angereist, eben gerade aus dem Hotel, direkt hierher gekommen. Perfekt, das ist ja witzig, so wie wir quasi. Wo kommt ihr denn her? Aus dem Schwarzwald. Aus dem Schwarzwald? Ja. Und was macht ihr hier? Urlaub. Urlaub. Eine Woche lang. Cool. Ich bin aus der Nähe von Nürnberg. Aus der Nähe von Amberg. Wir machen hier eine Woche Urlaub. Und wir waren gerade auf der Suche nach einem Picknickplatz. Wir kommen aus Berlin. Und sind Sie jetzt zum 1. Mai in New York? Ich ja, aber mein Mann war schon häufiger hier. Und seid ihr zum ersten Mal hier? Nee, also ich war hier abher. Du hast hier gelebt? Ja. Wir sind jetzt die vierte Woche hier. Also New York erst ein paar Tage, aber wir sind schon seit fast vier Wochen in Amerika. Wir haben Tagesplanung gemacht, haben uns gut überlegt, was wir uns anschauen wollen, um möglichst viel in den Tag zu kriegen und die Zeit auch sofort zu nutzen. Also seid ihr so richtig Touristen, die so abhaken, das, das, das muss alles angucken werden? So sieht es aus. Wie man sieht, bin ich von klein auf New York, seitdem ich denken kann. Wir haben hier kurz eine Präsentation gemacht und haben Matthias getroffen. Matthias, woher kommst du? Ich komme aus Österreich und den Niederlanden. Also mein Vater ist Österreicher, meine Mutter ist Holländisch. Wie ist es in New York zu leben? Was magst du an New York? Also in New York zu leben ist schon sehr spannend. Ich meine jeden Tag denke ich, oh ich lebe in New York. Das ist doch wirklich toll, um hier zu leben. Und es ist hektisch, es ist manchmal stressvoll. Also manchmal denke ich mir, oh ich würde am liebsten in einem kleinen Dorf in Österreich sein. Aber ja, in New York ist immer, immer was zu tun und immer was zu erleben. Was ist denn der größte Unterschied für dich zwischen Österreichern und Amerikanern? Oh, also die Österreicher, die sind in der Regel nicht so offen oder freundlich wie die Amerikaner. Ich weiß nicht, bei den Amerikanern ist es manchmal schwierig zu sehen, ob das jetzt wirklich ein echtes Freundlichsein ist oder eher oberflächlich. Aber generell ist es angenehmer, um mit Amerikanern umzugehen, vor allem mit den, die man nicht kennt. Also es ist vielleicht leichter, um Leute kennenzulernen in Amerika jedenfalls. Ja Leute, jetzt sind wir auf der Staten Island Ferry und bei mir sind Bob und Easy. Hallo. Hallo. Bob ist einer unserer Zuschauer und wohnt hier in New York. Im Juni hat er uns in Berlin besucht und als er gehört hat, dass wir in die USA kommen, hat er uns eingeladen. Ja, das stimmt. Und das ist der Wahnsinn, weil Bob ist tatsächlich für fünf Tage aus seinem eigenen Apartment ausgezogen, damit wir dort wohnen können. Bob, da möchten wir die Gelegenheit nutzen einfach mal Danke zu sagen. Danke Bob. Willkommen zu Hause. Ja, wir haben uns jetzt hier im Central Park mit Petrina Engelke getroffen und sie ist Bloggerin und Autorin aus Deutschland, wohnt aber in New York. Petrina, erstmal, warum wohnst du in New York? Wegen New York. Das alles? Ich habe nirgendwo mehr Geschichten gefunden als in dieser Stadt und da ich vom Geschichten erzählen lebe, war das irgendwie ein logischer Schritt an mir hinzuziehen. Ist New York typisch amerikanisch? Das glaube ich nicht. Also ich habe viel mehr Zeit in New York verbracht als in anderen Gegenden in Amerika. Ich habe auch nie anderswo als in New York und in Deutschland gelebt. Insofern weiß ich nicht, ob ich die beste Expertin für diese Frage ist, aber ich habe den Eindruck, New York ist anders. Das ist so ein bisschen wie, der Vatikan ist nicht Italien oder das wäre jetzt ein ganz krasses Beispiel. Aber New York ist schon sehr speziell, was ich zum Teil auf die Insellage schiebe. Warum? |

|  |
| --- |
| Musik Herzlich willkommen zurück bei Easy German. Wir sind heute in New York. Wie ihr wisst, machen wir eine vierwöchige USA-Reise. Und das ist unsere erste Station. Bei uns sind Easy. Hi Easy! Hallo! Ben? Hallo! Unser Übersetzer und Mara. Hi Mara! Hi! Woher kommst du? Ich komme aus Pennsylvania, aber ich studiere an der Uni in New York. Wie kommst du, dass du so gut Deutsch sprichst? Ich lerne Deutsch seit vier Jahren. Sehr gut. Und wir vier gehen heute mal durch New York und suchen dabei Deutsche. Wir wollen Interviews machen, erfahren, was die Menschen hier machen. Mal sehen, wen wir auf dem Weg treffen. Los geht's! Wir haben jetzt gerade vier Deutsche getroffen am Times Square. Sie haben sich dadurch verraten, dass sie Deutsch gesprochen haben. Wo kommt ihr denn her? Nähe Bielefeld. Und wie lange seid ihr schon in den USA? Gestern angereist, eben gerade aus dem Hotel, direkt hierher gekommen. Perfekt, das ist ja witzig, so wie wir quasi. Wo kommt ihr denn her? Aus dem Schwarzwald. Aus dem Schwarzwald? Ja. Und was macht ihr hier? Urlaub. Urlaub. Eine Woche lang. Cool. Ich bin aus der Nähe von Nürnberg. Aus der Nähe von Amberg. Wir machen hier eine Woche Urlaub. Und wir waren gerade auf der Suche nach einem Picknickplatz. Wir kommen aus Berlin. Und sind Sie jetzt zum 1. Mai in New York? Ich ja, aber mein Mann war schon häufiger hier. Und seid ihr zum ersten Mal hier? Nee, also ich war hier abher. Du hast hier gelebt? Ja. Wir sind jetzt die vierte Woche hier. Also New York erst ein paar Tage, aber wir sind schon seit fast vier Wochen in Amerika. Wir haben Tagesplanung gemacht, haben uns gut überlegt, was wir uns anschauen wollen, um möglichst viel in den Tag zu kriegen und die Zeit auch sofort zu nutzen. Also seid ihr so richtig Touristen, die so abhaken, das, das, das muss alles angucken werden? So sieht es aus. Wie man sieht, bin ich von klein auf New York, seitdem ich denken kann. Wir haben hier kurz eine Präsentation gemacht und haben Matthias getroffen. Matthias, woher kommst du? Ich komme aus Österreich und den Niederlanden. Also mein Vater ist Österreicher, meine Mutter ist Holländisch. Wie ist es in New York zu leben? Was magst du an New York? Also in New York zu leben ist schon sehr spannend. Ich meine jeden Tag denke ich, oh ich lebe in New York. Das ist doch wirklich toll, um hier zu leben. Und es ist hektisch, es ist manchmal stressvoll. Also manchmal denke ich mir, oh ich würde am liebsten in einem kleinen Dorf in Österreich sein. Aber ja, in New York ist immer, immer was zu tun und immer was zu erleben. Was ist denn der größte Unterschied für dich zwischen Österreichern und Amerikanern? Oh, also die Österreicher, die sind in der Regel nicht so offen oder freundlich wie die Amerikaner. Ich weiß nicht, bei den Amerikanern ist es manchmal schwierig zu sehen, ob das jetzt wirklich ein echtes Freundlichsein ist oder eher oberflächlich. Aber generell ist es angenehmer, um mit Amerikanern umzugehen, vor allem mit den, die man nicht kennt. Also es ist vielleicht leichter, um Leute kennenzulernen in Amerika jedenfalls. Ja Leute, jetzt sind wir auf der Staten Island Ferry und bei mir sind Bob und Easy. Hallo. Hallo. Bob ist einer unserer Zuschauer und wohnt hier in New York. Im Juni hat er uns in Berlin besucht und als er gehört hat, dass wir in die USA kommen, hat er uns eingeladen. Ja, das stimmt. Und das ist der Wahnsinn, weil Bob ist tatsächlich für fünf Tage aus seinem eigenen Apartment ausgezogen, damit wir dort wohnen können. Bob, da möchten wir die Gelegenheit nutzen einfach mal Danke zu sagen. Danke Bob. Willkommen zu Hause. Ja, wir haben uns jetzt hier im Central Park mit Petrina Engelke getroffen und sie ist Bloggerin und Autorin aus Deutschland, wohnt aber in New York. Petrina, erstmal, warum wohnst du in New York? Wegen New York. Das alles? Ich habe nirgendwo mehr Geschichten gefunden als in dieser Stadt und da ich vom Geschichten erzählen lebe, war das irgendwie ein logischer Schritt an mir hinzuziehen. Ist New York typisch amerikanisch? Das glaube ich nicht. Also ich habe viel mehr Zeit in New York verbracht als in anderen Gegenden in Amerika. Ich habe auch nie anderswo als in New York und in Deutschland gelebt. Insofern weiß ich nicht, ob ich die beste Expertin für diese Frage ist, aber ich habe den Eindruck, New York ist anders. Das ist so ein bisschen wie, der Vatikan ist nicht Italien oder das wäre jetzt ein ganz krasses Beispiel. Aber New York ist schon sehr speziell, was ich zum Teil auf die Insellage schiebe. Warum? |

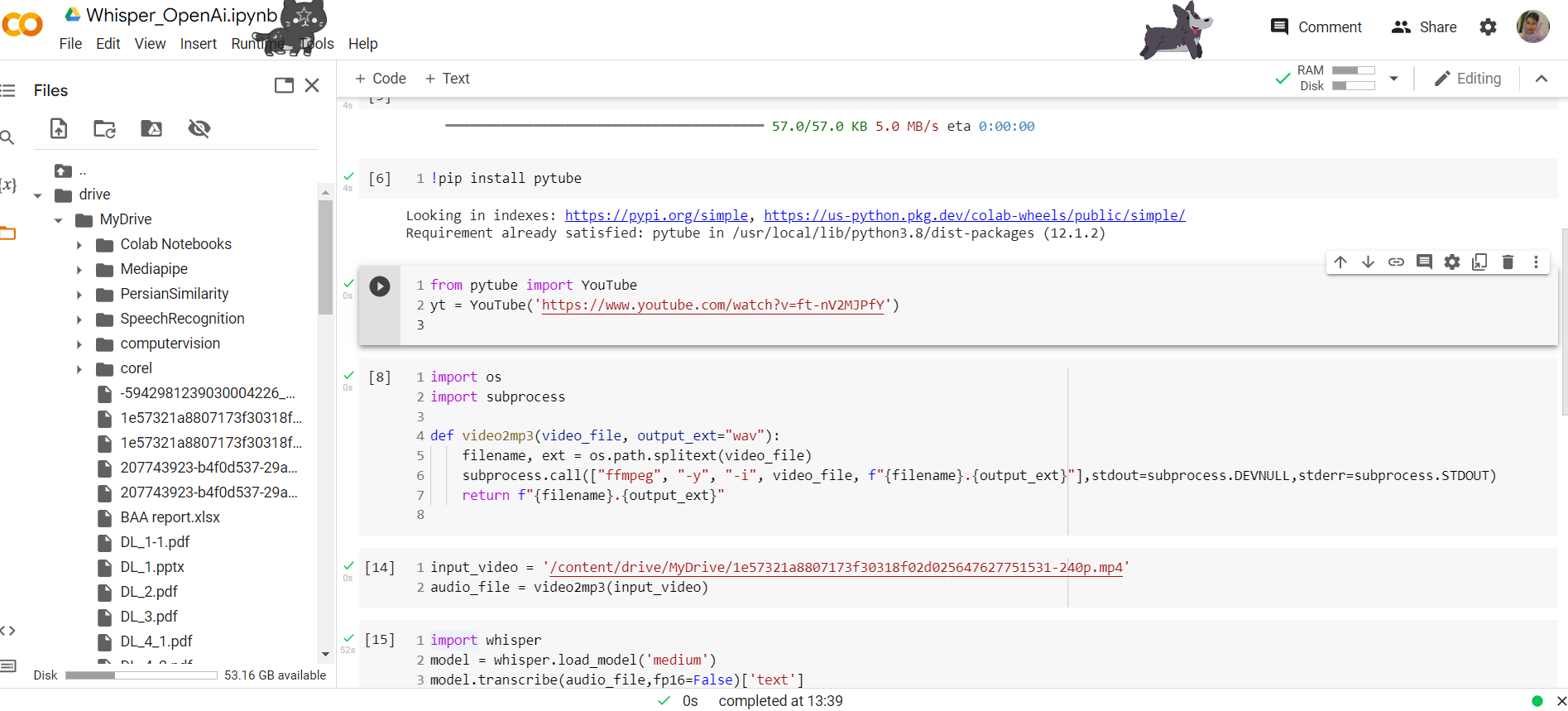
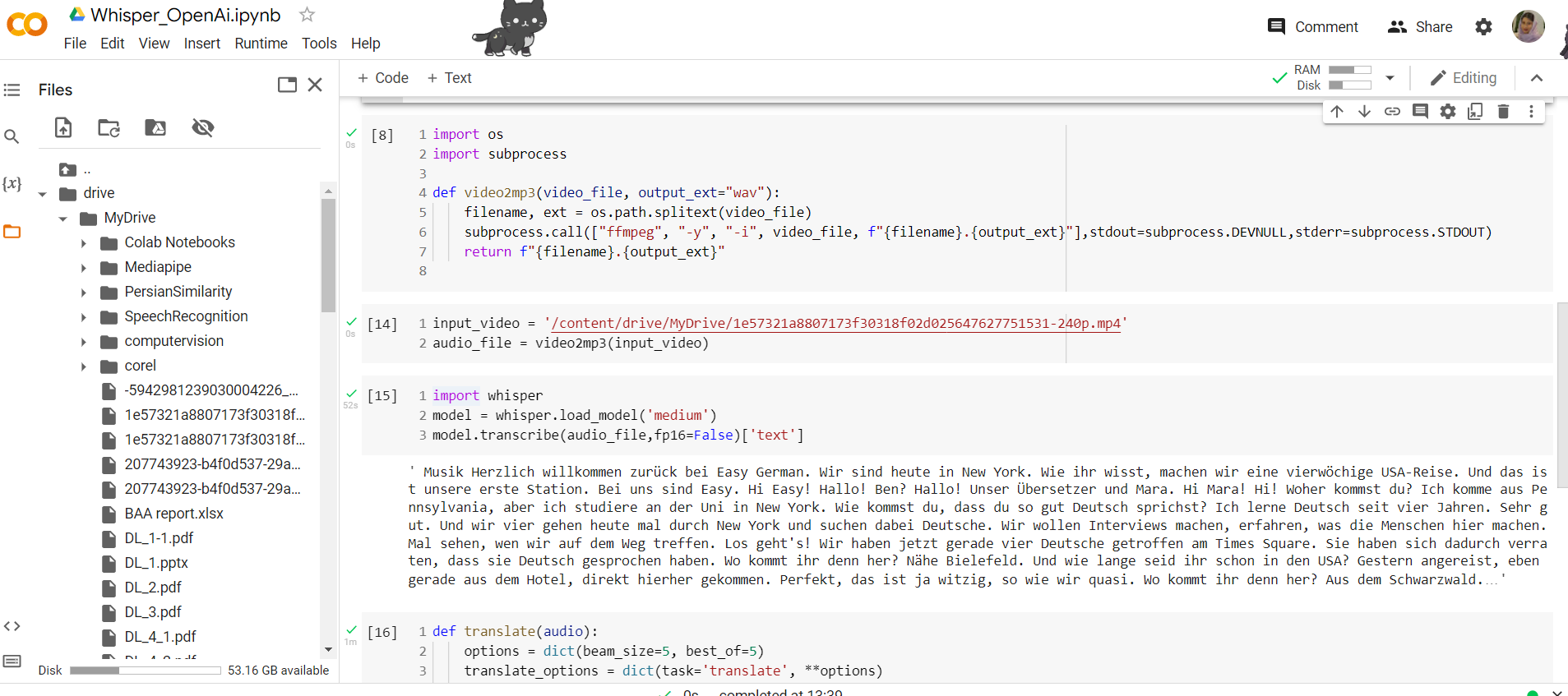
|  |
| --- |
| Musik Herzlich willkommen zurück bei Easy German. Wir sind heute in New York. Wie ihr wisst, machen wir eine vierwöchige USA-Reise. Und das ist unsere erste Station. Bei uns sind Easy. Hi Easy! Hallo! Ben? Hallo! Unser Übersetzer und Mara. Hi Mara! Hi! Woher kommst du? Ich komme aus Pennsylvania, aber ich studiere an der Uni in New York. Wie kommst du, dass du so gut Deutsch sprichst? Ich lerne Deutsch seit vier Jahren. Sehr gut. Und wir vier gehen heute mal durch New York und suchen dabei Deutsche. Wir wollen Interviews machen, erfahren, was die Menschen hier machen. Mal sehen, wen wir auf dem Weg treffen. Los geht's! Wir haben jetzt gerade vier Deutsche getroffen am Times Square. Sie haben sich dadurch verraten, dass sie Deutsch gesprochen haben. Wo kommt ihr denn her? Nähe Bielefeld. Und wie lange seid ihr schon in den USA? Gestern angereist, eben gerade aus dem Hotel, direkt hierher gekommen. Perfekt, das ist ja witzig, so wie wir quasi. Wo kommt ihr denn her? Aus dem Schwarzwald. Aus dem Schwarzwald? Ja. Und was macht ihr hier? Urlaub. Urlaub. Eine Woche lang. Cool. Ich bin aus der Nähe von Nürnberg. Aus der Nähe von Amberg. Wir machen hier eine Woche Urlaub. Und wir waren gerade auf der Suche nach einem Picknickplatz. Wir kommen aus Berlin. Und sind Sie jetzt zum 1. Mai in New York? Ich ja, aber mein Mann war schon häufiger hier. Und seid ihr zum ersten Mal hier? Nee, also ich war hier abher. Du hast hier gelebt? Ja. Wir sind jetzt die vierte Woche hier. Also New York erst ein paar Tage, aber wir sind schon seit fast vier Wochen in Amerika. Wir haben Tagesplanung gemacht, haben uns gut überlegt, was wir uns anschauen wollen, um möglichst viel in den Tag zu kriegen und die Zeit auch sofort zu nutzen. Also seid ihr so richtig Touristen, die so abhaken, das, das, das muss alles angucken werden? So sieht es aus. Wie man sieht, bin ich von klein auf New York, seitdem ich denken kann. Wir haben hier kurz eine Präsentation gemacht und haben Matthias getroffen. Matthias, woher kommst du? Ich komme aus Österreich und den Niederlanden. Also mein Vater ist Österreicher, meine Mutter ist Holländisch. Wie ist es in New York zu leben? Was magst du an New York? Also in New York zu leben ist schon sehr spannend. Ich meine jeden Tag denke ich, oh ich lebe in New York. Das ist doch wirklich toll, um hier zu leben. Und es ist hektisch, es ist manchmal stressvoll. Also manchmal denke ich mir, oh ich würde am liebsten in einem kleinen Dorf in Österreich sein. Aber ja, in New York ist immer, immer was zu tun und immer was zu erleben. Was ist denn der größte Unterschied für dich zwischen Österreichern und Amerikanern? Oh, also die Österreicher, die sind in der Regel nicht so offen oder freundlich wie die Amerikaner. Ich weiß nicht, bei den Amerikanern ist es manchmal schwierig zu sehen, ob das jetzt wirklich ein echtes Freundlichsein ist oder eher oberflächlich. Aber generell ist es angenehmer, um mit Amerikanern umzugehen, vor allem mit den, die man nicht kennt. Also es ist vielleicht leichter, um Leute kennenzulernen in Amerika jedenfalls. Ja Leute, jetzt sind wir auf der Staten Island Ferry und bei mir sind Bob und Easy. Hallo. Hallo. Bob ist einer unserer Zuschauer und wohnt hier in New York. Im Juni hat er uns in Berlin besucht und als er gehört hat, dass wir in die USA kommen, hat er uns eingeladen. Ja, das stimmt. Und das ist der Wahnsinn, weil Bob ist tatsächlich für fünf Tage aus seinem eigenen Apartment ausgezogen, damit wir dort wohnen können. Bob, da möchten wir die Gelegenheit nutzen einfach mal Danke zu sagen. Danke Bob. Willkommen zu Hause. Ja, wir haben uns jetzt hier im Central Park mit Petrina Engelke getroffen und sie ist Bloggerin und Autorin aus Deutschland, wohnt aber in New York. Petrina, erstmal, warum wohnst du in New York? Wegen New York. Das alles? Ich habe nirgendwo mehr Geschichten gefunden als in dieser Stadt und da ich vom Geschichten erzählen lebe, war das irgendwie ein logischer Schritt an mir hinzuziehen. Ist New York typisch amerikanisch? Das glaube ich nicht. Also ich habe viel mehr Zeit in New York verbracht als in anderen Gegenden in Amerika. Ich habe auch nie anderswo als in New York und in Deutschland gelebt. Insofern weiß ich nicht, ob ich die beste Expertin für diese Frage ist, aber ich habe den Eindruck, New York ist anders. Das ist so ein bisschen wie, der Vatikan ist nicht Italien oder das wäre jetzt ein ganz krasses Beispiel. Aber New York ist schon sehr speziell, was ich zum Teil auf die Insellage schiebe. Warum? |

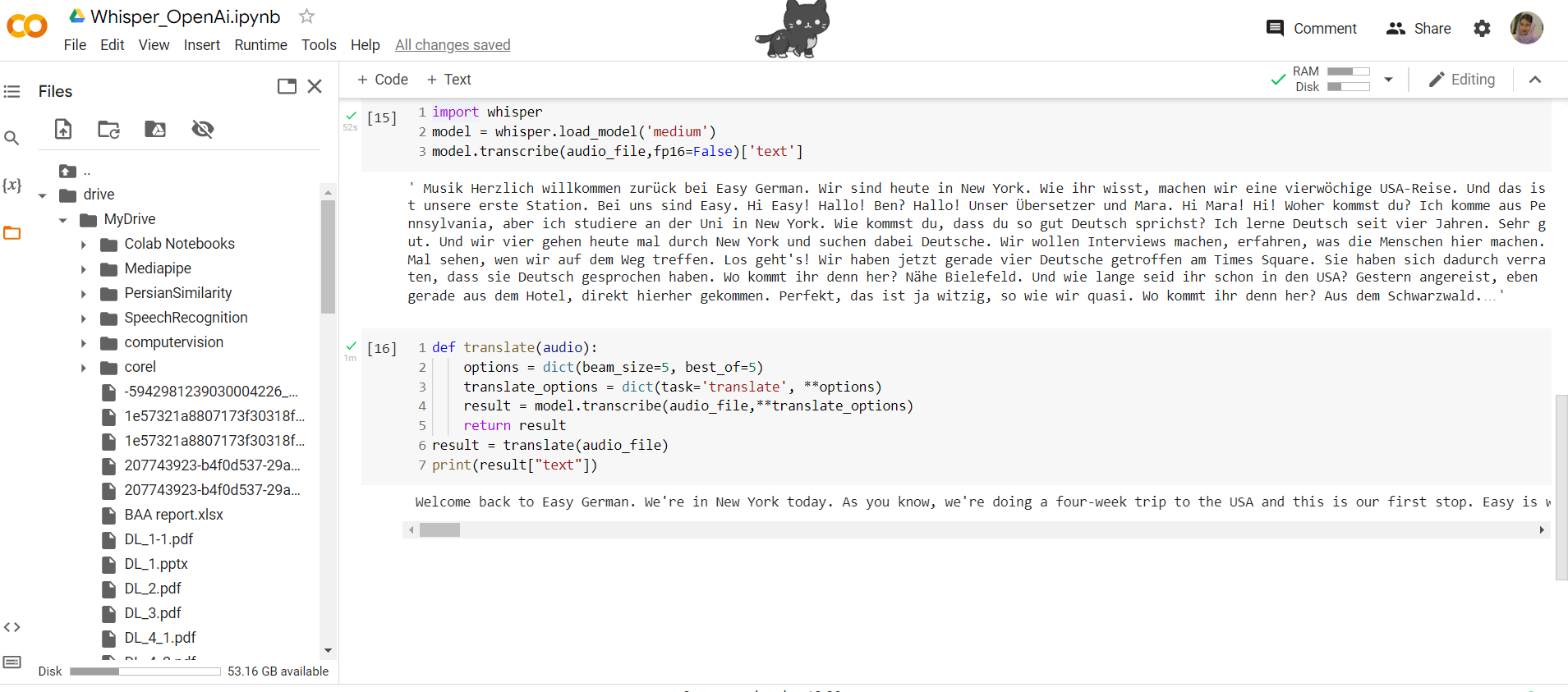
در مرحله ی بعد تابع translate را می نویسیم که متنی که از تبدیل گفتار به متن بدست آمده را با متدbeam search به زبان مقصد(انگلیسی) ترجمه کند .

|  |
| --- |
| def translate(audio):     options = dict(beam\_size=5, best\_of=5)     translate\_options = dict(task='translate', \*\*options)     result = model.transcribe(audio\_file,\*\*translate\_options)     return result result = translate(audio\_file) print(result["text"]) |

**خروجی**

|  |
| --- |
| Welcome back to Easy German. We're in New York today. As you know, we're doing a four-week trip to the USA and this is our first stop. Easy is with us. Hi Easy. Hello. Ben. Hello. Our translator and Mara. Hi Mara. Hi. Where are you from? I'm from Pennsylvania, but I'm studying at the university in New York. And how come you speak German so well? I've been learning German for four years. Very good. And we four, we're going through New York today and looking for Germans. We want to do interviews, find out what people are doing here. Let's see if we can meet on the way. Let's go. Yes, we just met four Germans here at the Times Square. They revealed that they spoke German. Where are you from? Near Bielefeld. Near Bielefeld. And how long have you been in the USA? We arrived yesterday. We just came straight from the hotel. Perfect. That's funny. Just like us. Where are you from? From the Black Forest. Black Forest? Yes. What are you doing here? Vacation. Vacation. For a week. Yes. Cool. Yes, I'm from near Nuremberg. From near Amberg. We're on vacation here for a week. And yes, we were just on the way to visit a picnic place. We come from Berlin. And are you in New York for the first time? I am, but my husband was here more often. And are you here for the first time? No, I was here before. Ah, you lived here? Yes. We're here for the fourth week now. So New York, only a few days, but we've been in America for almost four weeks. We made a daily schedule, thought about what we want to look at to get as many people in the day as possible and to make the most of the time. So you're really tourists who park like that. All that has to be looked at. That's what it looks like. As you can see, I've been in New York since I can think. We made a short presentation here and met Matthias. Matthias, where are you from? I'm from Austria and the Netherlands. So my father is Austrian and my mother is Dutch. How is it to live in New York? What do you like about New York? Well, living in New York is very exciting. I mean, every day I think, oh, I live in New York. That's really great to live here. And it's hectic. It's sometimes stressful. So sometimes I think, oh, I'd prefer to be in a small village in Austria. But yes, in New York there is always something to do and always something to experience. What is the biggest difference for you between Austrians and Americans? Well, the Austrians are usually not as open or friendly as the Americans. I don't know, for Americans it's sometimes difficult to see if it's a real friendliness or more superficial. But generally it's more pleasant to hang out with Americans, especially with those you don't know. So it's maybe easier to get to know people in America anyway. Yes, guys, now we are on the Staten Island ferry and with me are Bob and Easy. Hello. Hello. Bob is one of our viewers and lives here in New York. In June he visited us in Berlin and when he heard that we were coming to the USA, he invited us. Yes, that's true. And that's amazing because Bob actually moved out of his own apartment for five days so that we could live there. Bob, we would like to take this opportunity to say thank you. Welcome home. Yes, we have now met here in Central Park with Petrina Engelke and she is a blogger and author from Germany but lives in New York. Petrina, first of all, why do you live in New York? Because of New York. That's all? I have found more stories than in this city and since I live from storytelling, it was somehow a logical step to take here. Is New York typically American? I don't think so. I have spent much more time in New York than in other areas in America. I have never lived anywhere else than in New York and Germany. In this respect, I don't know if I am the best expert on this question but I have the impression that New York is different. The Vatican is not Italy. That would be a very bad example. But New York is very special, which I partially push back to the island situation. Why? |

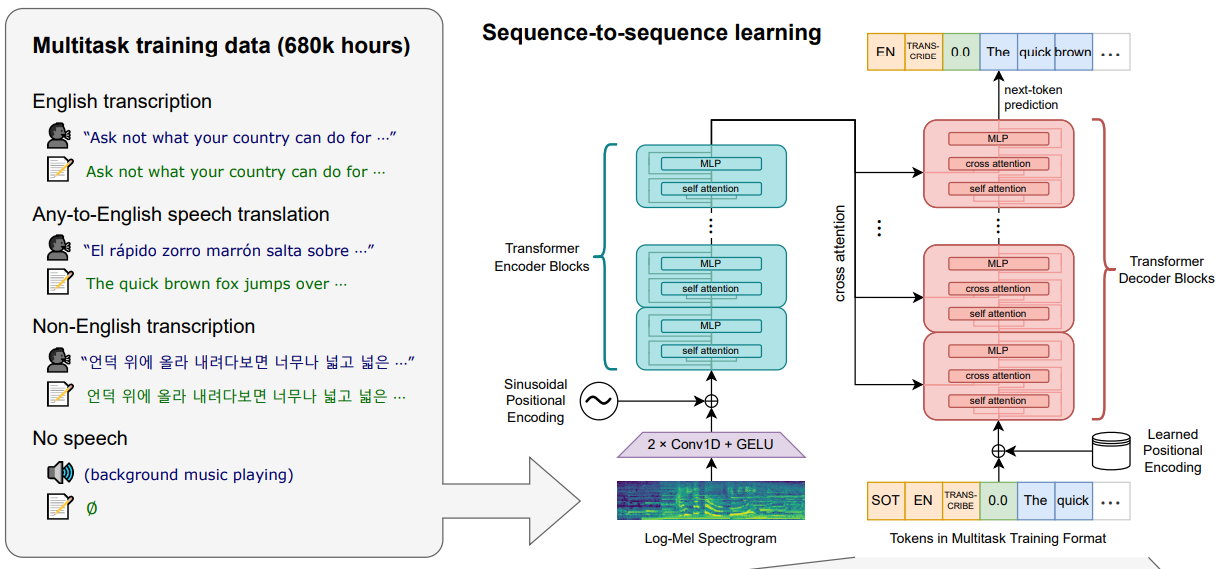
******اجرا در محیط کولب**



**معماری مدل**

معماری Whisper یک رویکرد ساده از end-to-end است که به عنوان یکencoder-decoder transformerپیاده سازی شده است. صدای ورودی به تکه‌های 30 ثانیه‌ای تقسیم می‌شود، به یک spectrogram log-Mel تبدیل می‌شود و سپس به یک encoder ارسال می‌شود. یک encoder برای پیش‌بینی عنوان متن مربوطه، با token‌های خاصی ترکیب شده است که مدل واحد را برای انجام وظایفی مانند شناسایی زبان، timestamp در سطح عبارت، transcribe گفتار چند زبانه و ترجمه گفتار به انگلیسی تبدیل می‌کند.

هدف بررسی قابلیت‌های supervised-pre-trainedدر مقیاس بزرگ برای تشخیص گفتار است، از یک معماری off-the-self استفاده می‌شود تا یافته‌های خود را با پیشرفت‌های مدل اشتباه نگیریم. یک encoder-decoder transformer را انتخاب می کنیم .تمام صداها مجدداً تا 16000 هرتز نمونه‌برداری می‌شوند و یک نمایش spectogram Mel 80کانالی بر روی پنجره‌های 25 میلی‌ثانیه‌ای با stride 10 میلی‌ثانیه محاسبه می‌شود. برای normalization ویژگی‌ها، ورودی را بین -1 و 1 scale می‌کنیم. میانگین تقریباً صفر در مجموعه داده های قبل از آموزش است. Theencoder نمایش ورودی را با یک stem کوچک متشکل از دو لایه convolution با عرض فیلتر 3 و تابع فعال‌سازی GELU پردازش می‌کند (که در آن لایه کانولوشن دوم دارای stride دو است. سپس positional embeddingسینوسی به خروجی stem اضافه می‌شوند و پس از آن بلوک‌های encoder transformer اعمال می‌شوند. transformer از residual block از قبل فعال سازی استفاده می کند و یک نرمال سازی لایه نهایی به خروجی encoder اعمال می شود. encoder از positional embedding های آموخته شده و نمایش token های ورودی-خروجی استفاده می کند encoderو decoder دارای عرض و تعداد بلوک های ترانسفورماتور یکسان هستند.

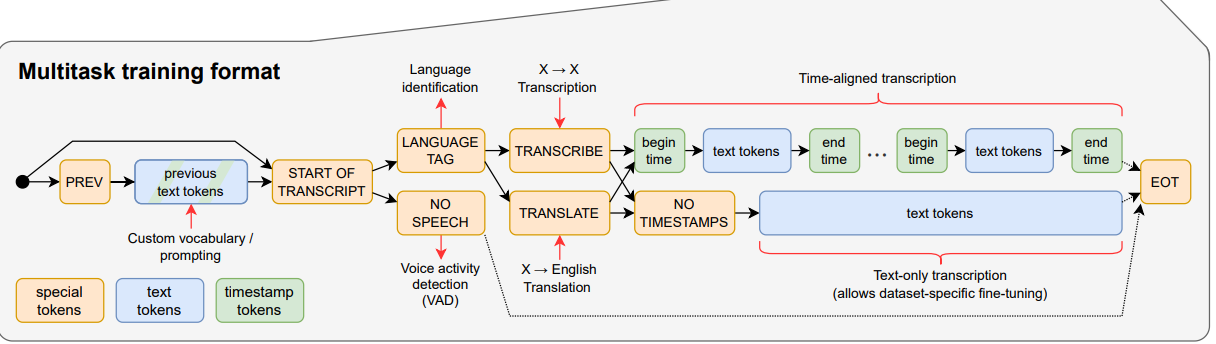


**Multitask format**

یک سیستم تشخیص گفتار با ویژگی‌های کامل می‌تواند شامل بسیاری از مؤلفه‌های اضافی مانند تشخیص فعالیت صوتی، reverse text normalization باشد. این مولفه ها اغلب به طور جداگانه مدیریت می شوند و در نتیجه یک سیستم نسبتاً پیچیده در اطراف مدل اصلی تشخیص گفتار ایجاد می شود. برای کاهش این پیچیدگی، درکل یک مدل واحد pipeline پردازش گفتار را انجام می دهیم، نه فقط بخش تشخیص هسته. یک نکته مهم در اینجا رابط مدل است.

decoder یک مدل زبان شرطی صوتی است، مدل آموزش داده می شود تا متن transcript را شرطی کند، و از یاد بگیرد از متن متنی با برد بلندتر برای حل صدای مبهم استفاده کند. به طور خاص، با مقداری احتمال، متن transcript قبل از بخش صوتی فعلی را به زمینه decoder اضافه می شود.

ابتدا، ما زبان مورد صحبت را پیش‌بینی می‌کنیم که با یک token منحصر به فرد برای هر زبان در مجموعه آموزشی ما نشان داده می‌شود. در مواردی که هیچ گفتاری در یک بخش صوتی وجود ندارد، مدل برای پیش‌بینی نشانه <nospeech|> آموزش داده می‌شود که این را نشان می‌دهد. token بعدی وظیفه (اعم از transcript یا ترجمه) را با یکtoken<|transcript|> یا <|translate|> مشخص می کند. پس از این، مشخص می‌کنیم که آیا باید timestamp را پیش‌بینی کنیم یا نه با اضافه کردن یک token <notimestamps|برای آن مورد. در این مرحله وظیفه و فرمت مورد نظر به طور کامل مشخص شده و خروجی شروع می شود. برای پیش‌بینی timestamp، زمان را نسبت به بخش صوتی کنونی پیش‌بینی می‌کنیم که همه زمان‌ها را با دقت 20 میلی‌ثانیه تعیین می‌کند که با وضوح زمانی مدل‌های Whisper مطابقت دارد، و برای هر یک از آنها token‌های اضافی به واژگان خود اضافه می‌کنیم.

token زمان شروع قبل از متن هر transcript پیش‌بینی می‌شود و token زمان پایان بعد از آن پیش‌بینی می‌شود.

**قطعه کد کار با whisperX**

این مدل timestamp های مدل Whisper OpenAI را از طریق force alignment با مدل‌های ASR مبتنی بر واج مانند (wav2vec2.0)، که مورد استفاده در multilingual هسند را ، اصلاح می‌کند.

Whisper در حالی که transcript های بسیار دقیقی تولید می کند، timestamp های مربوطه در expression-level هستند، نه در هر کلمه، و ممکن است در چند ثانیه نادرست باشند.

ASR phoneme based مجموعه‌ای از مدل‌های fine-tuneبرای تشخیص کوچک‌ترین واحد گفتاری که یک کلمه را از کلمه دیگر متمایز می‌کند است، به عنوان مثال. عنصر p در "tap". یک مدل نمونه wav2vec2.0 است.

Force alignment به فرآیندی اشاره دارد که طی آنtranscript املایی با ضبط‌های صوتی align می‌شوند تا به‌طور خودکار segmentation phone level تولید شود**.**

**موارد جدید اضافه شده نسبت به مدل بیس whisper**

فیلتر Voice Activity Detection (VAD) از Pyannote.audio به عنوان یک مرحله پیش پردازش برای حذف اتکا به whisper time stamps و تنها transcript بخش‌های صوتی حاوی گفتار استفاده می‌شود. flag --vad\_filter را اضافه کنید، دقت timestamps و robustness را افزایش می دهد (به دلیل ورودی های 30 ثانیه در wav2vec2 به حافظه گرافیکی بیشتری نیاز دارد)

Character level timestamps (خروجی فایل char.ass )

Diarization) ) هنوز در بتا است، --diarize را اضافه می کنیم)

برای فعال کردن فیلتر VAD و Diarization، توکن Hugging Face access خود را اضافه می کنیم از اینجا پس از آرگومان --hf\_token ایجاد می کنیم.

**مجموعه ی داده ی مورد نیاز برای پروژه** **whisper**

مجموعه ی داده ی common voice برای زبان فارسی

برای زبان های غیر انگلیسی از جمله فارسیFleurs مجموعه داده

چهار ویس از مکالمات به زبان های انگلیسی و آلمانی و فرانسوی و ژاپنی که در پوشه ی examples است

**قطعه کد مدل custom شده برای whisperX**

ابتدا کتابخانه های مورد نیاز و مدل بیس را نصب می کنیم . FFMPEG هم برای فشرده سازی و تبدیل ویدیو به گفتار ضروری است

|  |
| --- |
| pip install -e . |

|  |
| --- |
| !pip install git+https://github.com/openai/whisper.git |

|  |
| --- |
| !sudo apt update && sudo apt install ffmpeg |

حالا مجموعه ی داده sample01.wav که شامل یک مکالمه انگلیسی هست را transcribe می کنیم .

|  |
| --- |
| !whisperx examples/sample01.wav |

**خروجی برای زبان انگلیسی**

|  |
| --- |
| 100%|████████████████████████████████████████| 461M/461M [00:02<00:00, 165MiB/s] Downloading: "https://download.pytorch.org/torchaudio/models/wav2vec2\_fairseq\_base\_ls960\_asr\_ls960.pth" to /root/.cache/torch/hub/checkpoints/wav2vec2\_fairseq\_base\_ls960\_asr\_ls960.pth 100% 360M/360M [00:01<00:00, 250MB/s] Performing transcription... Detecting language using up to the first 30 seconds. Use `--language` to specify the language Detected language: English [00:00.000 --> 00:03.000]  Ella, how are you? [00:03.000 --> 00:09.000]  Oh, I'm OK. I will be. [00:09.000 --> 00:12.000]  I said she could stay with us, Mum. I just thought she feels better. [00:12.000 --> 00:13.000]  Of course she can. [00:13.000 --> 00:15.000]  Oh, this won't be for long. [00:15.000 --> 00:17.000]  Well, you can stay as long as you want, my love. [00:17.000 --> 00:19.000]  Oh, you really missed you. [00:19.000 --> 00:20.000]  Pops. [00:20.000 --> 00:21.000]  Great to see you, love. [00:21.000 --> 00:22.000]  Oh. [00:22.000 --> 00:24.000]  All right, shall we get you off to bed, then? [00:24.000 --> 00:26.000]  You should have given me some room [00:26.000 --> 00:28.000]  and I'd have put the electric blanket on. [00:28.000 --> 00:29.200]  All right, then. [00:29.200 --> 00:31.000]  Freezing up there. [00:31.000 --> 00:34.000]  In a bedroom, Peter unpacks her suitcase. [00:34.000 --> 00:38.000]  The middle-aged woman opens her green case. [00:38.000 --> 00:40.000]  Do you want, yeah, PJs? [00:40.000 --> 00:42.000]  Yeah. [00:42.000 --> 00:44.000]  Lifting a bundle of pajamas, [00:44.000 --> 00:46.000]  Peter finds a sheet of paper labeled [00:46.000 --> 00:50.000]  Lancaster North Hospital Discharged Sheet. [00:50.000 --> 00:53.000]  He closes the suitcase and brings glory to the pajamas. [00:53.000 --> 00:54.000]  There you go. [00:54.000 --> 00:55.000]  Thank you. [00:55.000 --> 00:57.000]  He picks up the locket. [00:57.000 --> 00:58.680]  You kept it. [00:58.680 --> 01:00.680]  Oh, of course. Performing alignment... |

**خروجی مدل بر روی یک مجموعه ی داده ی medium**

|  |
| --- |
| !whisperx examples/sample01.wav --model medium.en --vad\_filter --align\_model WAV2VEC2\_ASR\_LARGE\_LV60K\_960H |

|  |
| --- |
| Downloading (...)lve/main/config.yaml: 100% 277/277 [00:00<00:00, 46.7kB/s] Downloading (...)"pytorch\_model.bin";: 100% 17.7M/17.7M [00:00<00:00, 123MB/s] Downloading (...)eech2021/config.yaml: 100% 1.98k/1.98k [00:00<00:00, 932kB/s] 100%|██████████████████████████████████████| 72.1M/72.1M [00:00<00:00, 170MiB/s] Downloading: "https://download.pytorch.org/torchaudio/models/wav2vec2\_fairseq\_large\_lv60k\_asr\_ls960.pth" to /root/.cache/torch/hub/checkpoints/wav2vec2\_fairseq\_large\_lv60k\_asr\_ls960.pth 100% 1.18G/1.18G [00:20<00:00, 63.0MB/s] Performing VAD... ~~ Transcribing VAD chunk: (00:00.000 --> 00:29.219) ~~ [00:00.000 --> 00:01.000]  Hello. [00:01.000 --> 00:02.000]  Hello. [00:02.000 --> 00:03.000]  Hello. [00:03.000 --> 00:04.000]  Hello. [00:04.000 --> 00:05.000]  Hello. [00:05.000 --> 00:06.000]  Hello. [00:06.000 --> 00:07.000]  Hello. [00:07.000 --> 00:08.000]  Hi. [00:08.000 --> 00:09.000]  Okay. [00:09.000 --> 00:10.000]  I will be. [00:10.000 --> 00:11.000]  I said she could stay with us, Ma. [00:11.000 --> 00:12.000]  I just thought she feels better. [00:12.000 --> 00:13.000]  Oh, she can. [00:13.000 --> 00:14.000]  Oh, she can't. [00:14.000 --> 00:15.000]  She's won't be for long. [00:15.000 --> 00:17.000]  Well, you can stay as long as you want. [00:17.000 --> 00:18.000]  My love. [00:18.000 --> 00:20.000]  Really missed you. [00:20.000 --> 00:21.000]  Perhaps. [00:21.000 --> 00:22.000]  Great to see you. [00:22.000 --> 00:23.000]  Oh. [00:23.000 --> 00:25.000]  And I sure we get you off to bed then. [00:25.000 --> 00:26.000]  You should take me some warm. [00:26.000 --> 00:28.000]  I was about electric blankets, aren't we? [00:28.000 --> 00:33.000]  What type of bike do I put in there? ~~ Transcribing VAD chunk: (00:29.624 --> 00:59.054) ~~ [00:00.000 --> 00:01.840]  Please, I'm up there. [00:01.840 --> 00:04.640]  In a bedroom, Peter unpacks her suitcase. [00:04.640 --> 00:08.560]  The middle-aged woman opens her green case. [00:08.560 --> 00:09.280]  Do you want air? [00:09.280 --> 00:10.240]  T.J.'s. [00:10.240 --> 00:12.760]  Yeah. [00:12.760 --> 00:14.720]  Lifting a bundle of pajamas, Peter [00:14.720 --> 00:18.600]  finds a sheet of paper labeled Lancaster North Hospital. [00:18.600 --> 00:20.680]  Discharge sheet. [00:20.680 --> 00:23.280]  He closes the suitcase and brings glory of the pajamas. [00:23.280 --> 00:24.640]  Here you go. [00:24.640 --> 00:26.040]  Thank you. [00:26.040 --> 00:27.520]  He picks up the lock it. [00:27.520 --> 00:29.160]  You kept it. [00:29.160 --> 00:30.000]  Oh. ~~ Transcribing VAD chunk: (00:59.797 --> 00:59.965) ~~ Performing alignment... |

**خروجی برای زبان فارسی**

ابتدا با استفاده از hugging face دیتاست common voice را دانلود می کنیم

دیتاست common voice از یک MP3 unique و فایل متنی متناظر تشکیل شده اس آوریمت. بسیاری از 9283 ساعت ثبت‌شده در مجموعه داده‌ها همچنین شامل demographic metadata مانند سن، جنس و لهجه هستند که می‌توانند به آموزش دقت موتورهای تشخیص گفتار کمک کنند. مجموعه داده در حال حاضر شامل 7335 ساعت معتبر به 60 زبان از جمله زبان فارسی است

|  |
| --- |
| !pip install datasets |

|  |
| --- |
| from datasets import list\_datasets, load\_dataset from pprint import pprint  datasets\_list = list\_datasets()  pprint(datasets\_list,compact=True) |

چون دیتاست حجم بالایی دارد فقط قسمت زبان فارسی آن را دانلود می کنیم

|  |
| --- |
| dataset = load\_dataset('common\_voice','fa') |

طول دیتاست و اولین آیتم از مجموعه ی داده ی آموزشی را بدست می آوریم

|  |
| --- |
| print(f" Dataset len(dataset): {len(dataset)}") print("\n First item 'dataset[0]':") pprint(dataset['train']) |

|  |
| --- |
| Dataset len(dataset): 6  First item 'dataset[0]': Dataset({     features: ['client\_id', 'path', 'audio', 'sentence', 'up\_votes', 'down\_votes', 'age', 'gender', 'accent', 'locale', 'segment'],     num\_rows: 7593 }) |

نام و مسیر و نرخ نمونه برداری برای تعدادی از داده های آموزش جهت نمایش و transcribe صدا

|  |
| --- |
| # Or get slices with several examples: pprint(dataset['train']['audio']) |

|  |
| --- |
| 'sampling\_rate': 48000},  {'array': array([ 0.0000000e+00,  4.3126418e-14, -1.8672672e-16, ...,         3.3252991e-07,  4.9133985e-07,  8.3890455e-08], dtype=float32),   'path': '/root/.cache/huggingface/datasets/downloads/extracted/3b0e0ce56990e1c035c009dd3032760da75614498aab0a4090ea91c8d4e8a843/cv-corpus-6.1-2020-12-11/fa/clips/common\_voice\_fa\_23713726.mp3',   'sampling\_rate': 48000},  {'array': array([ 0.0000000e+00, -3.3857930e-14, -1.7956780e-14, ...,         5.1054503e-06, -1.6895583e-06, -3.0443393e-06], dtype=float32),   'path': '/root/.cache/huggingface/datasets/downloads/extracted/3b0e0ce56990e1c035c009dd3032760da75614498aab0a4090ea91c8d4e8a843/cv-corpus-6.1-2020-12-11/fa/clips/common\_voice\_fa\_23713727.mp3',   'sampling\_rate': 48000},  {'array': array([ 0.0000000e+00, -3.0736535e-13, -1.6106154e-13, ...,         1.2761238e-06,  1.5162623e-06,  4.4175795e-07], dtype=float32),   'path': '/root/.cache/huggingface/datasets/downloads/extracted/3b0e0ce56990e1c035c009dd3032760da75614498aab0a4090ea91c8d4e8a843/cv-corpus-6.1-2020-12-11/fa/clips/common\_voice\_fa\_23713728.mp3',   'sampling\_rate': 48000},  {'array': array([0.00000000e+00, 3.59715716e-14, 1.03599665e-13, ...,        6.69242436e-05, 4.16793337e-05, 1.31940669e-05], dtype=float32),   'path': '/root/.cache/huggingface/datasets/downloads/extracted/3b0e0ce56990e1c035c009dd3032760da75614498aab0a4090ea91c8d4e8a843/cv-corpus-6.1-2020-12-11/fa/clips/common\_voice\_fa\_23713729.mp3',   'sampling\_rate': 48000}, |

**نمایش صدا**

|  |
| --- |
| from IPython.display import Audio, display  sound\_file='/root/.cache/huggingface/datasets/downloads/extracted/3b0e0ce56990e1c035c009dd3032760da75614498aab0a4090ea91c8d4e8a843/cv-corpus-6.1-2020-12-11/fa/clips/common\_voice\_fa\_24072987.mp3'  display(Audio(sound\_file, autoplay=True)) |

|  |
| --- |
| !whisperx  /root/.cache/huggingface/datasets/downloads/extracted/3b0e0ce56990e1c035c009dd3032760da75614498aab0a4090ea91c8d4e8a843/cv-corpus-6.1-2020-12-11/fa/clips/common\_voice\_fa\_24072987.mp3 |

**خروجی برای زبان فارسی دو صدای متفاوت از دیتاست**

|  |
| --- |
| Performing transcription... Detecting language using up to the first 30 seconds. Use `--language` to specify the language Detected language: Persian [00:00.000 --> 00:04.540]  این پتو در تا بسان خونک و در زمستان گرمست New language found (fa)! Previous was (en), loading new alignment model for new language... Performing alignment... |

|  |
| --- |
| !whisperx /root/.cache/huggingface/datasets/downloads/extracted/3b0e0ce56990e1c035c009dd3032760da75614498aab0a4090ea91c8d4e8a843/cv-corpus-6.1-2020-12-11/fa/clips/common\_voice\_fa\_24071976.mp3 |

|  |
| --- |
| Performing transcription... Detecting language using up to the first 30 seconds. Use `--language` to specify the language Detected language: Persian [00:00.000 --> 00:03.000]  ایی بابا تو دنبال درد سر میگردی New language found (fa)! Previous was (en), loading new alignment model for new language... Performing alignment... |

**زبان ایتالیایی بر روی مدل tiny**

|  |
| --- |
| !whisperx --model tiny --language it examples/sample\_it\_01.wav |

|  |
| --- |
| 100%|██████████████████████████████████████| 72.1M/72.1M [00:00<00:00, 154MiB/s] Downloading: "https://download.pytorch.org/torchaudio/models/wav2vec2\_voxpopuli\_base\_10k\_asr\_it.pt" to /root/.cache/torch/hub/checkpoints/wav2vec2\_voxpopuli\_base\_10k\_asr\_it.pt 100% 360M/360M [00:01<00:00, 240MB/s] Performing transcription... [00:00.000 --> 00:03.000]  Signori, piacciere. [00:03.000 --> 00:05.000]  Lì avvicite la vedeta amirata [00:05.000 --> 00:07.000]  Tattutino ecco di salgemo a provera [00:07.000 --> 00:09.000]  della nostra cultura seranno naturalmente [00:09.000 --> 00:10.000]  a coltisuto la mia protezione [00:10.000 --> 00:12.000]  per la durata del loro soggiardo. [00:15.000 --> 00:16.000]  Grazie. [00:17.000 --> 00:18.000]  Corlomi? [00:18.000 --> 00:20.000]  Lo pronuncio correttamente? [00:21.000 --> 00:22.000]  Si. [00:22.000 --> 00:23.000]  Greta. [00:23.000 --> 00:25.000]  Corlomi? [00:25.000 --> 00:28.000]  Per costessia me lo repete ancora. [00:28.000 --> 00:34.000]  Gordlami, scusico me, Gordlami. [00:34.000 --> 00:36.000]  Ancora una volta. [00:38.000 --> 00:40.000]  Gordlami. [00:40.000 --> 00:42.000]  Come si chiama lei? [00:42.000 --> 00:45.000]  Antonio Margaretti. [00:45.000 --> 00:47.000]  Ancora? [00:47.000 --> 00:49.000]  Margaretti. [00:49.000 --> 00:50.000]  Una altra volta. [00:50.000 --> 00:53.000]  Ma adesso vuole proprio se dire la musica delle parole. [00:53.000 --> 00:56.000]  Margaretti. [00:56.000 --> 01:03.000]  margariti e lei, domenicte coco come domenicte coco bravo bravo Performing alignment... |

**زبان آلمانی**

|  |
| --- |
| !whisperx --model tiny --language de examples/sample\_de\_01.wav |

|  |
| --- |
| Downloading: "https://download.pytorch.org/torchaudio/models/wav2vec2\_voxpopuli\_base\_10k\_asr\_de.pt" to /root/.cache/torch/hub/checkpoints/wav2vec2\_voxpopuli\_base\_10k\_asr\_de.pt 100% 360M/360M [00:05<00:00, 69.4MB/s] Performing transcription... [00:00.000 --> 00:02.000]  Weilerein von Hamas-Mann. [00:06.000 --> 00:08.000]  Wo war das Lande jetzt lange her? [00:09.000 --> 00:11.000]  Schneidig, eh und hier. [00:14.000 --> 00:16.000]  Also, was ist mit ihrem wunderschönen Bein geschehen? [00:17.000 --> 00:21.000]  Ein Nebenprodukt der Arsch dritte, die sind eine deutsche Filmwelt aus der Zweifelzone. [00:22.000 --> 00:24.000]  Sparen Sie sich regulante Sie, Alter Hund. [00:24.000 --> 00:27.000]  Ich kenne zu viele von ihren früheren Eroberungen, [00:27.000 --> 00:29.440]  dass ich in ihren Hohnichtopftreten könnte. [00:29.440 --> 00:31.040]  Nein, ich muss das bisschen. [00:31.760 --> 00:33.360]  Ja, ich habe mich, [00:33.680 --> 00:35.800]  du mal so muss ich eigentlich stehen, [00:36.680 --> 00:38.520]  im Bergsteigen versucht. [00:40.960 --> 00:43.120]  Bergsteigen? Da war ja uns im Beit verletzt, [00:43.120 --> 00:44.240]  bei Bergsteigen. [00:44.240 --> 00:45.620]  Ab这 istlerom, oder nicht? [00:47.460 --> 00:48.260]  Hahaha. Performing alignment... |

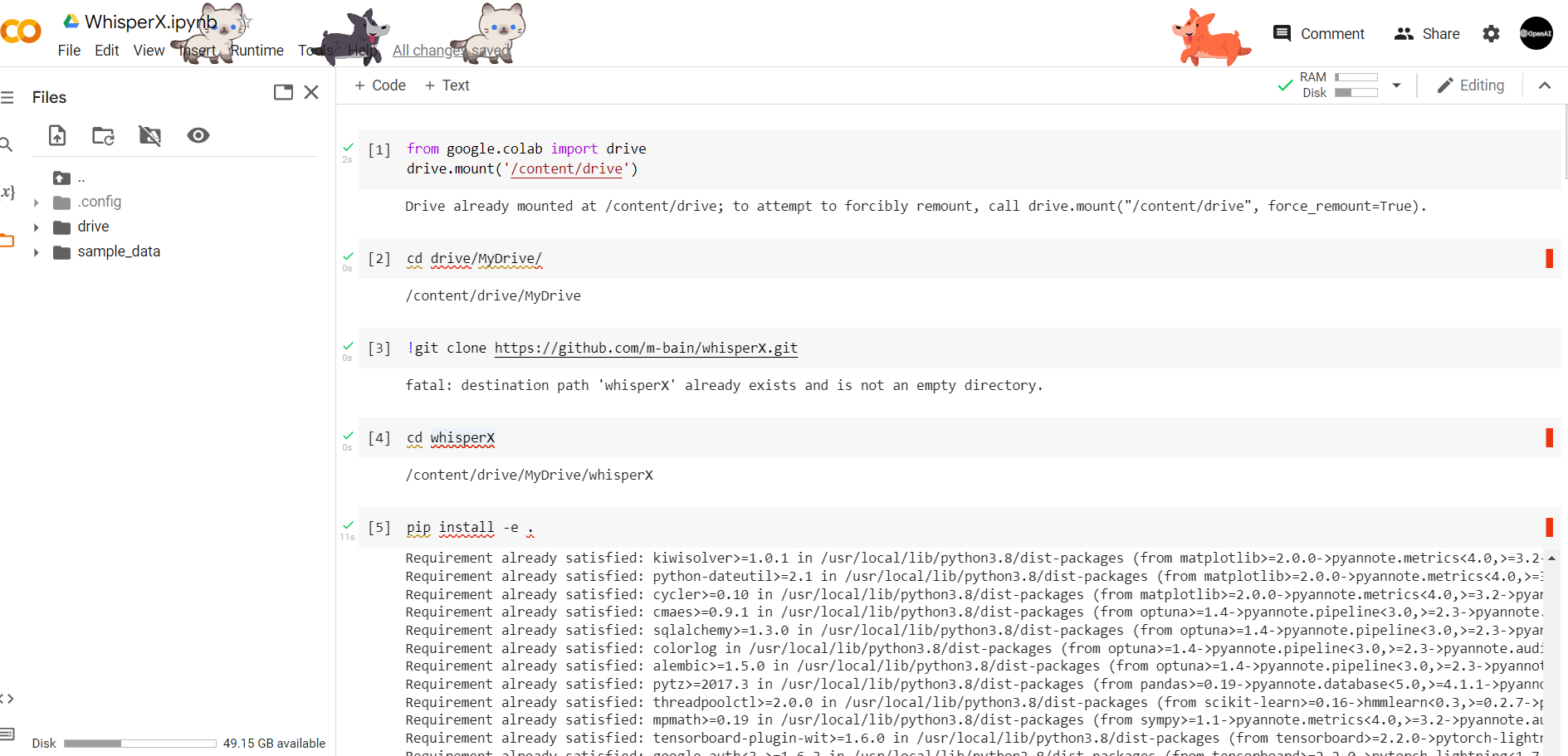
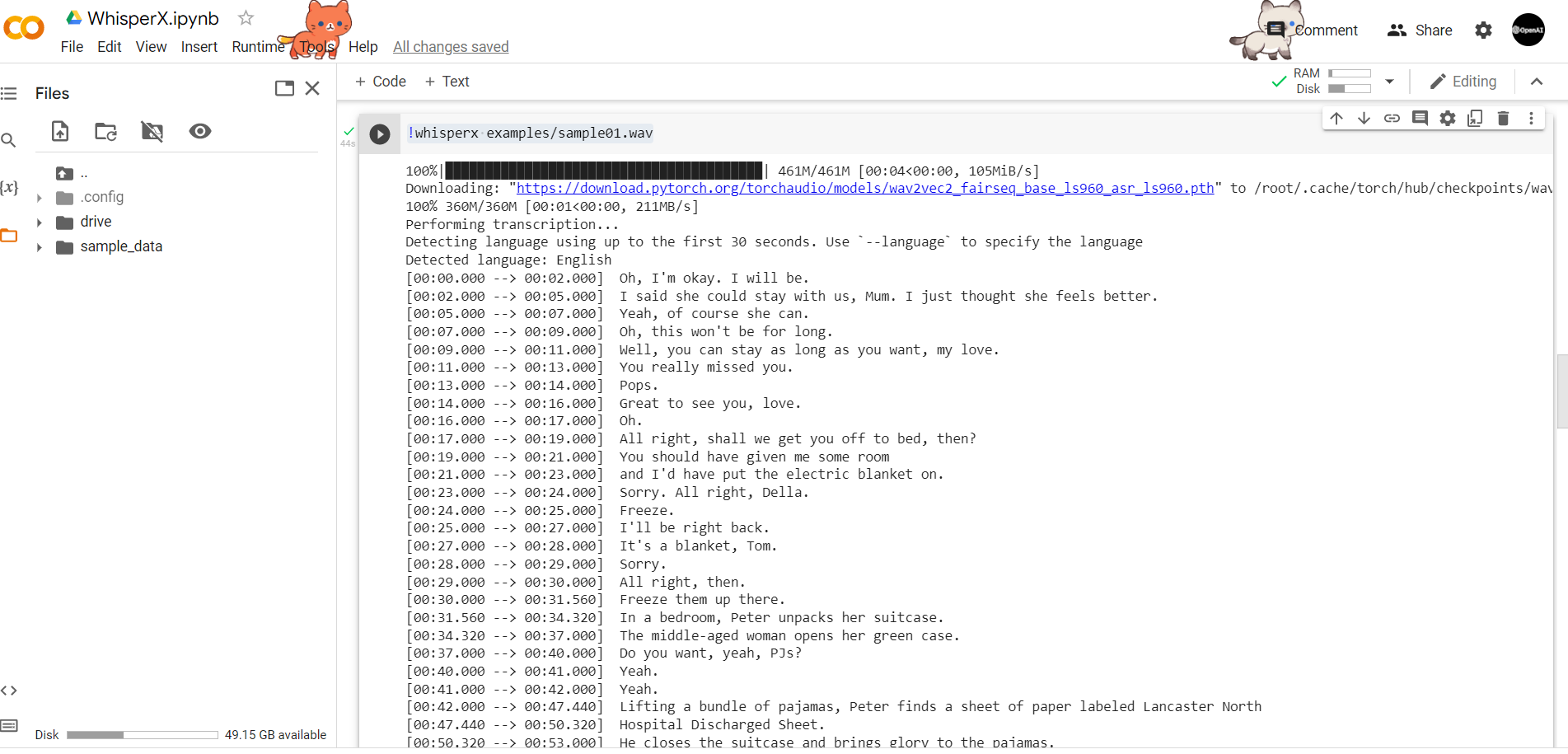
**دو ویدیو خروجی transcribe که توسط مدل آموزش داده شده اند در پوشه ی output قرار گرفته اند**

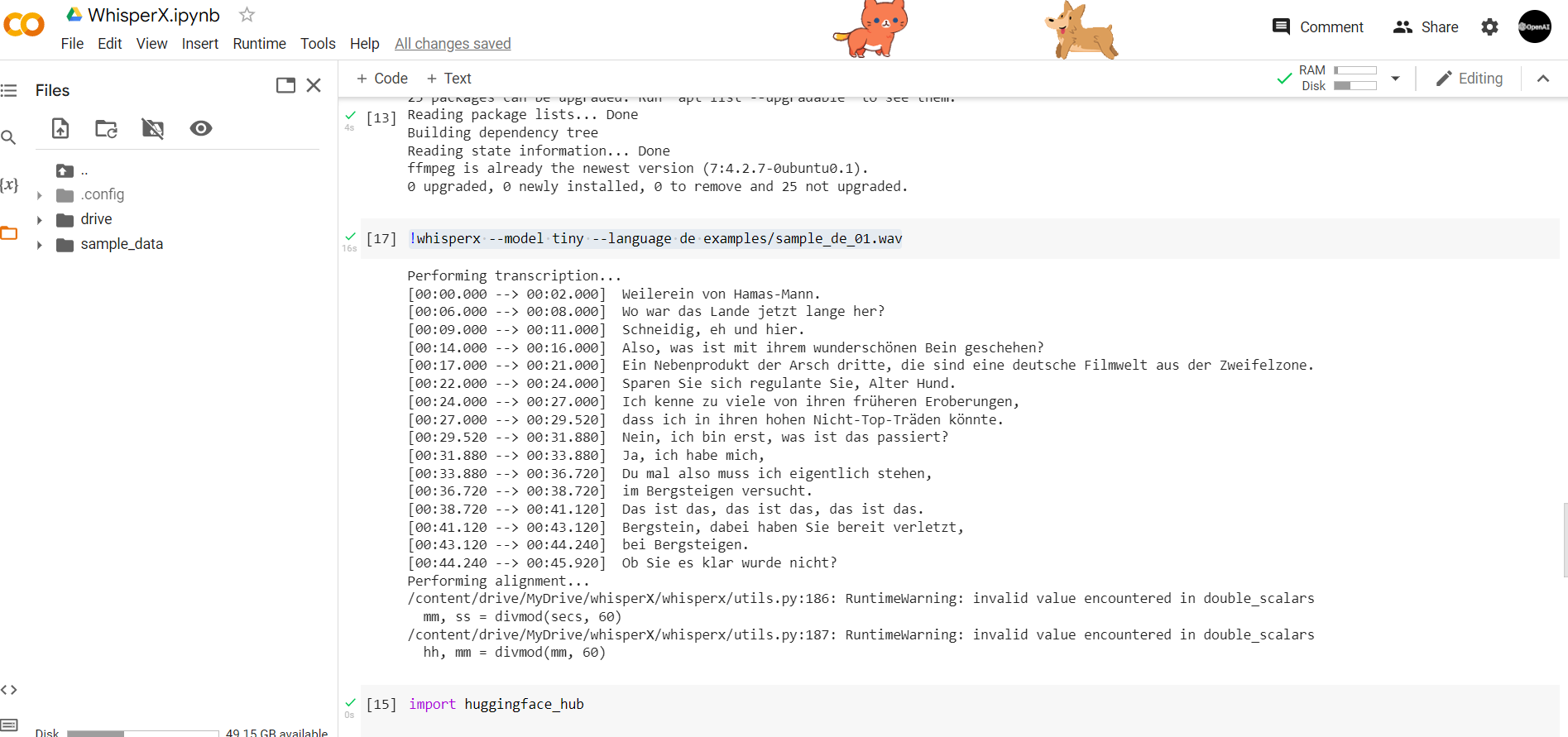
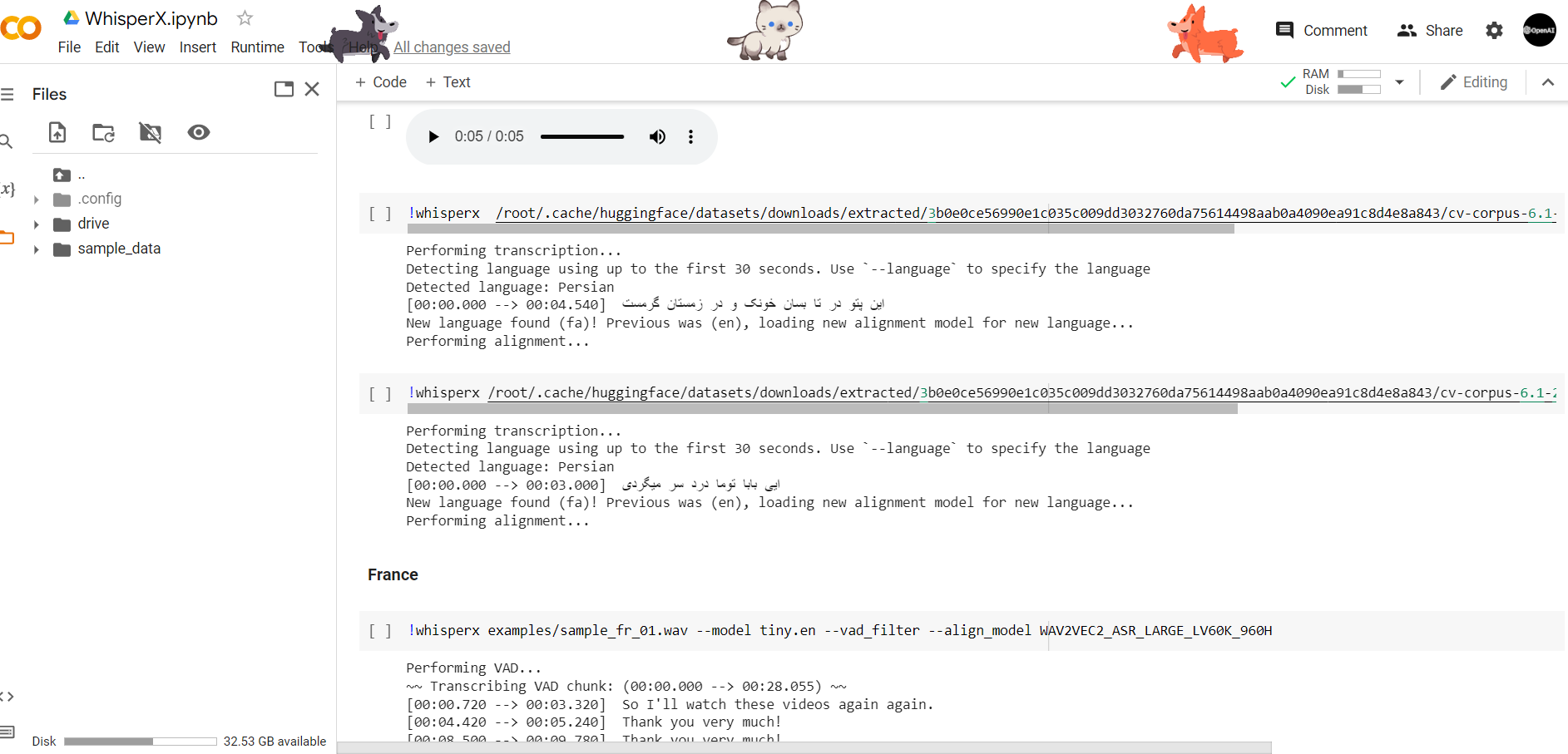
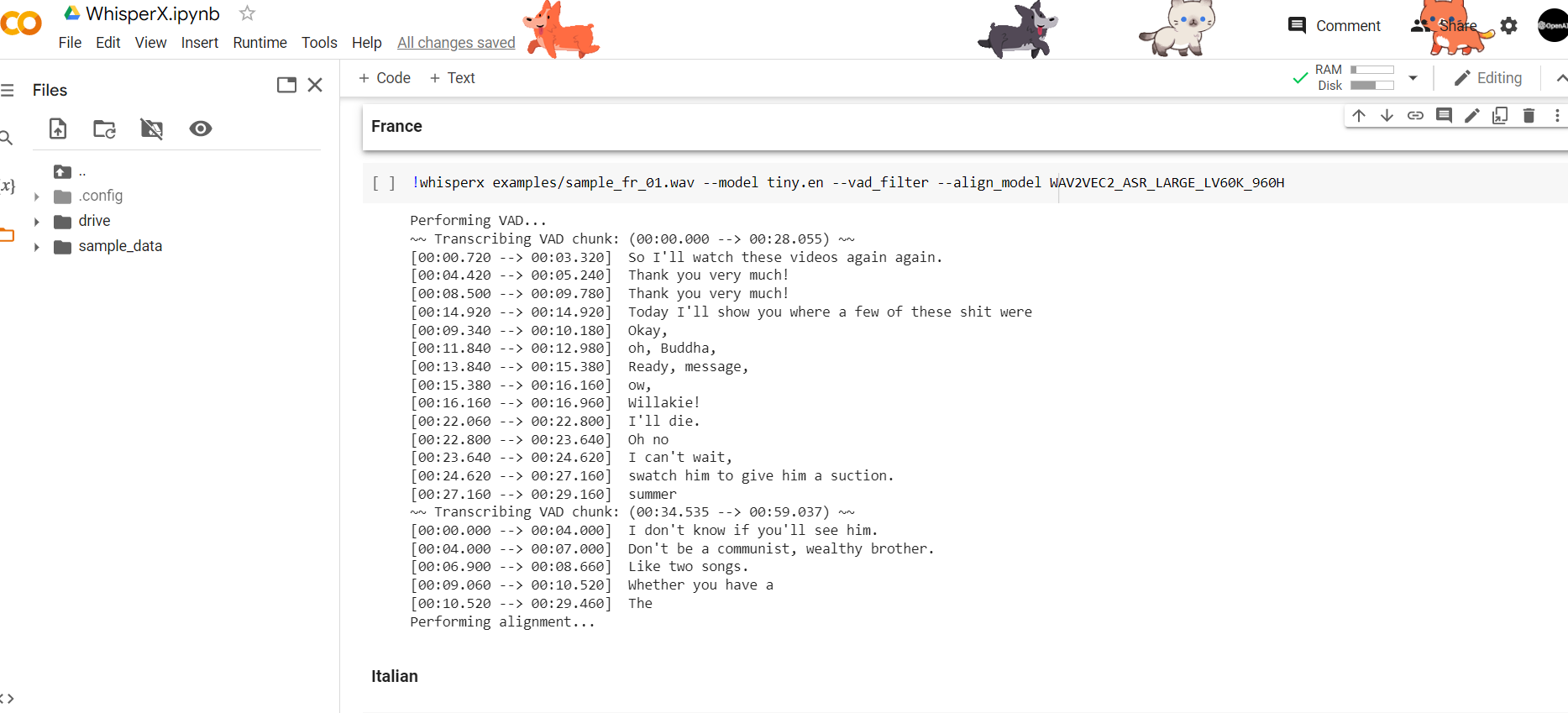
|  |
| --- |
| show video |

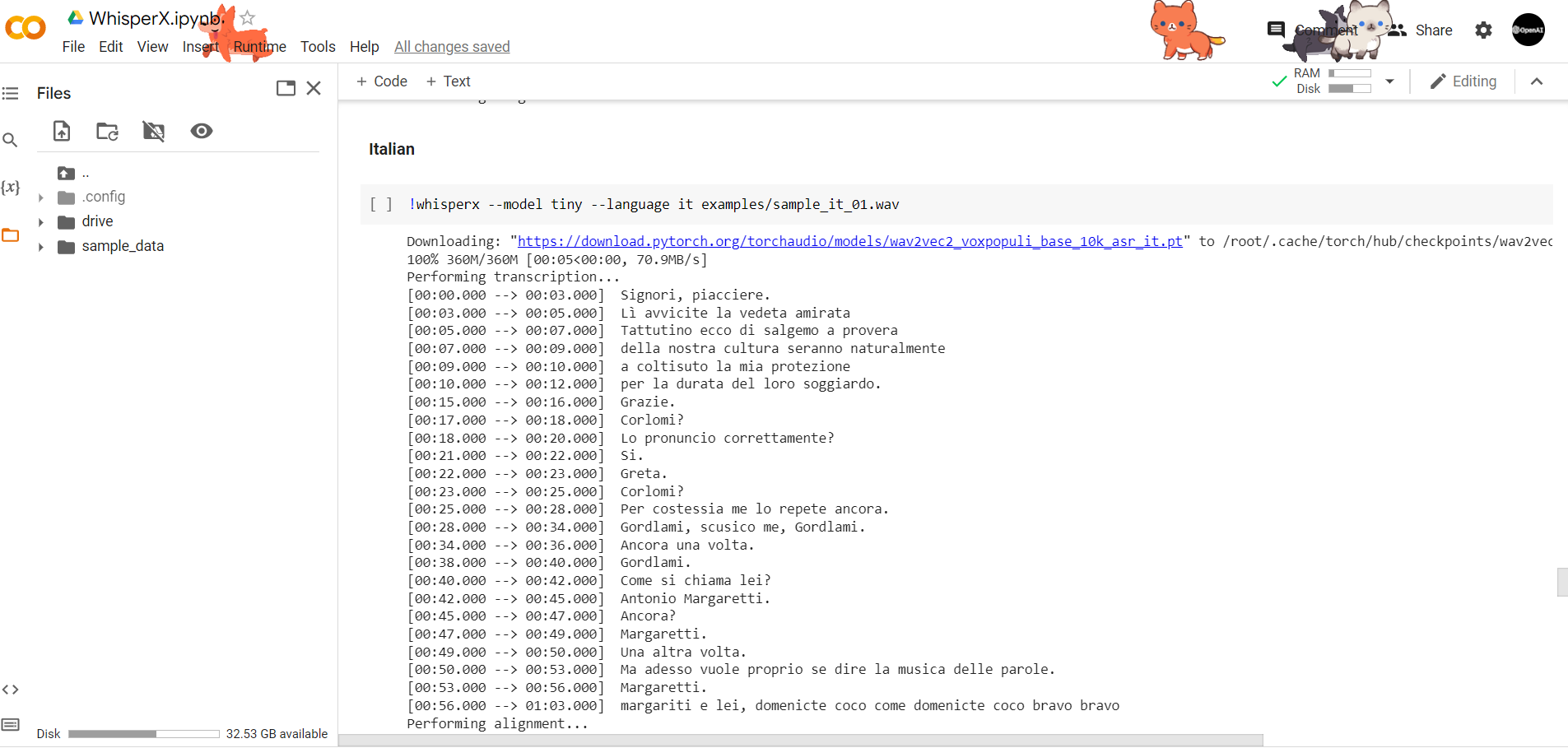
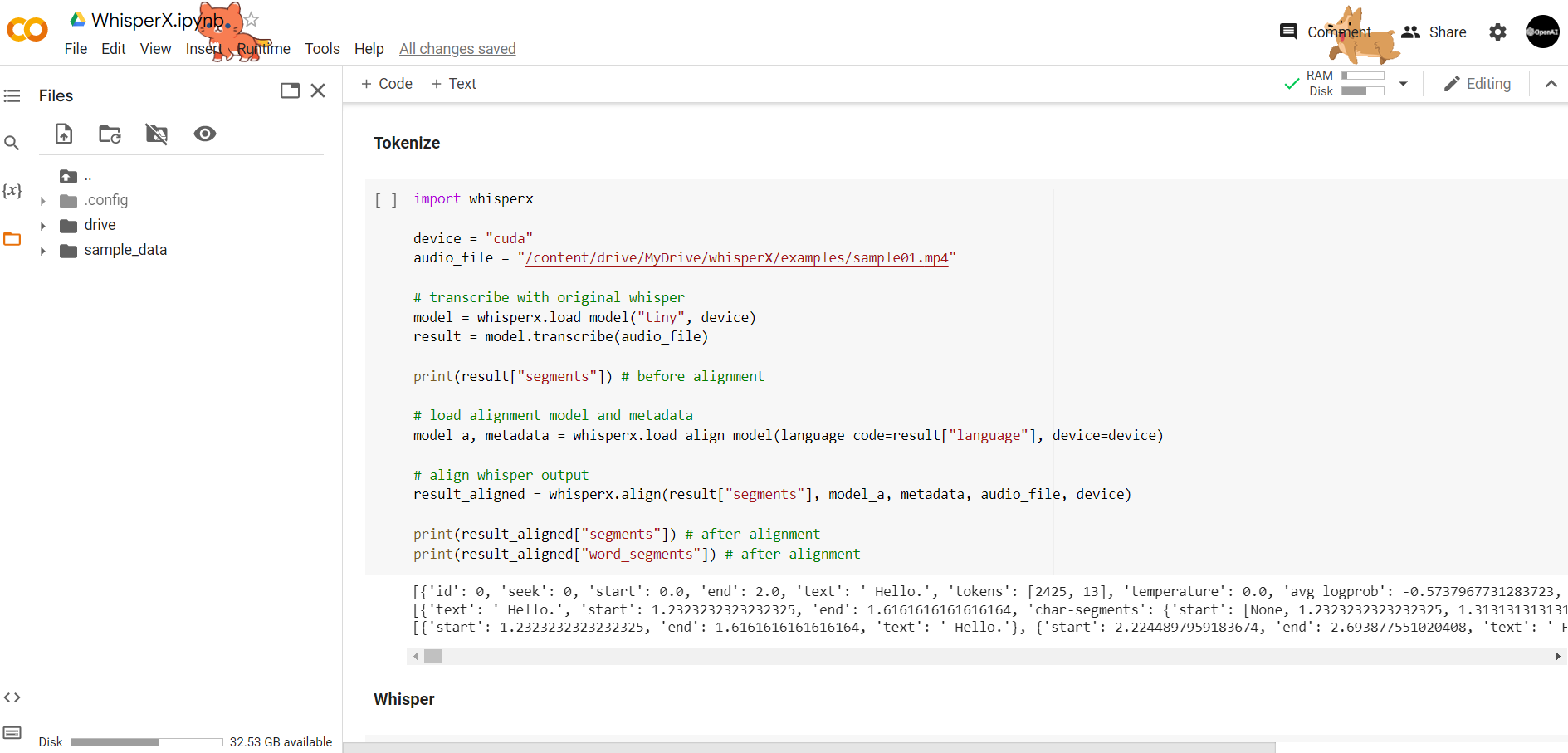
|  |
| --- |
| from IPython.display import HTML from base64 import b64encode mp4 = open('/content/drive/MyDrive/whisperX/examples/sample01.mp4','rb').read() data\_url = "data:/content/drive/MyDrive/whisperX/examples/sample01.mp4;base64," + b64encode(mp4).decode() HTML(""" <video width=400 controls>       <source src="%s" type="video/mp4"> </video> """ % data\_url) |

|  |
| --- |
| Tokenize |

|  |
| --- |
| import whisperx  device = "cuda"  audio\_file = "/content/drive/MyDrive/whisperX/examples/sample01.mp4"  # transcribe with original whisper model = whisperx.load\_model("tiny", device) result = model.transcribe(audio\_file)  print(result["segments"]) # before alignment  # load alignment model and metadata model\_a, metadata = whisperx.load\_align\_model(language\_code=result["language"], device=device)  # align whisper output result\_aligned = whisperx.align(result["segments"], model\_a, metadata, audio\_file, device)  print(result\_aligned["segments"]) # after alignment print(result\_aligned["word\_segments"]) # after alignment |

**اجرا در کولب**

****

**قطعه کد Multilingual ASR**

ابتدا کتابخانه های مهم را ایمپورت می کنیم

|  |
| --- |
| import io import os import numpy as np  try:     import tensorflow  # required in Colab to avoid protobuf compatibility issues except ImportError:     pass  import torch import pandas as pd import urllib import tarfile import whisper import torchaudio  from scipy.io import wavfile from tqdm.notebook import tqdm   pd.options.display.max\_rows = 100 pd.options.display.max\_colwidth = 1000 DEVICE = "cuda" if torch.cuda.is\_available() else "cpu" |

**Loading the Fleurs dataset**

زبان مجموعه داده فلور را برای دانلود انتخاب می کنیم. عملکرد transcript و ترجمه بسته به زبان بسیار متفاوت است. یک selection box در صفحه ایجاد میشود که ما زبان فارسی را انتخاب می کنیم.زبان هایی که به غیر از انگلیسی ساپورت می شوند در زیر نمایش داده شده است

|  |
| --- |
| import ipywidgets as widgets  languages = {"af\_za": "Afrikaans", "am\_et": "Amharic", "ar\_eg": "Arabic", "as\_in": "Assamese", "az\_az": "Azerbaijani", "be\_by": "Belarusian", "bg\_bg": "Bulgarian", "bn\_in": "Bengali", "bs\_ba": "Bosnian", "ca\_es": "Catalan", "cmn\_hans\_cn": "Chinese", "cs\_cz": "Czech", "cy\_gb": "Welsh", "da\_dk": "Danish", "de\_de": "German", "el\_gr": "Greek", "en\_us": "English", "es\_419": "Spanish", "et\_ee": "Estonian", "fa\_ir": "Persian", "fi\_fi": "Finnish", "fil\_ph": "Tagalog", "fr\_fr": "French", "gl\_es": "Galician", "gu\_in": "Gujarati", "ha\_ng": "Hausa", "he\_il": "Hebrew", "hi\_in": "Hindi", "hr\_hr": "Croatian", "hu\_hu": "Hungarian", "hy\_am": "Armenian", "id\_id": "Indonesian", "is\_is": "Icelandic", "it\_it": "Italian", "ja\_jp": "Japanese", "jv\_id": "Javanese", "ka\_ge": "Georgian", "kk\_kz": "Kazakh", "km\_kh": "Khmer", "kn\_in": "Kannada", "ko\_kr": "Korean", "lb\_lu": "Luxembourgish", "ln\_cd": "Lingala", "lo\_la": "Lao", "lt\_lt": "Lithuanian", "lv\_lv": "Latvian", "mi\_nz": "Maori", "mk\_mk": "Macedonian", "ml\_in": "Malayalam", "mn\_mn": "Mongolian", "mr\_in": "Marathi", "ms\_my": "Malay", "mt\_mt": "Maltese", "my\_mm": "Myanmar", "nb\_no": "Norwegian", "ne\_np": "Nepali", "nl\_nl": "Dutch", "oc\_fr": "Occitan", "pa\_in": "Punjabi", "pl\_pl": "Polish", "ps\_af": "Pashto", "pt\_br": "Portuguese", "ro\_ro": "Romanian", "ru\_ru": "Russian", "sd\_in": "Sindhi", "sk\_sk": "Slovak", "sl\_si": "Slovenian", "sn\_zw": "Shona", "so\_so": "Somali", "sr\_rs": "Serbian", "sv\_se": "Swedish", "sw\_ke": "Swahili", "ta\_in": "Tamil", "te\_in": "Telugu", "tg\_tj": "Tajik", "th\_th": "Thai", "tr\_tr": "Turkish", "uk\_ua": "Ukrainian", "ur\_pk": "Urdu", "uz\_uz": "Uzbek", "vi\_vn": "Vietnamese", "yo\_ng": "Yoruba"} selection = widgets.Dropdown(     options=[("Select language", None), ("----------", None)] + sorted([(f"{v} ({k})", k) for k, v in languages.items()]),     value="ko\_kr",     description='Language:',     disabled=False, )  selection |

|  |
| --- |
| lang = selection.value language = languages[lang]  assert lang is not None, "Please select a language" print(f"Selected language: {language} ({lang})") |

|  |
| --- |
| Selected language: Persian (fa\_ir) |

**بخش دانلود دیتاست**

با توجه به زبانی که در باکس بخش بالا انتخاب کردیم یک تابع برای دانلود دیتاست زبان مورد نظر می نویسیم که ازgoogleapi storage urlدیتاست فارسی فلور را دانلود کرده و در یک دایترکتوری لوکال ذخیره می کند. البته فایل هایی را می خواند که با پسوند wav باشند

|  |
| --- |
| def download(url: str, target\_path: str):     with urllib.request.urlopen(url) as source, open(target\_path, "wb") as output:         with tqdm(total=int(source.info().get("Content-Length")), ncols=80, unit='iB', unit\_scale=True, unit\_divisor=1024) as loop:             while True:                 buffer = source.read(8192)                 if not buffer:                     break                  output.write(buffer)                 loop.update(len(buffer))   class Fleurs(torch.utils.data.Dataset):     """     A simple class to wrap Fleurs and subsample a portion of the dataset as needed.     """     def \_\_init\_\_(self, lang, split="test", subsample\_rate=1, device=DEVICE):         url = f"https://storage.googleapis.com/xtreme\_translations/FLEURS102/{lang}.tar.gz"         tar\_path = os.path.expanduser(f"~/.cache/fleurs/{lang}.tgz")         os.makedirs(os.path.dirname(tar\_path), exist\_ok=True)          if not os.path.exists(tar\_path):             download(url, tar\_path)          all\_audio = {}         with tarfile.open(tar\_path, "r:gz") as tar:             for member in tar.getmembers():                 name = member.name                 if name.endswith(f"{split}.tsv"):                     labels = pd.read\_table(tar.extractfile(member), names=("id", "file\_name", "raw\_transcription", "transcription", "\_", "num\_samples", "gender"))                  if f"/{split}/" in name and name.endswith(".wav"):                     audio\_bytes = tar.extractfile(member).read()                     all\_audio[os.path.basename(name)] = wavfile.read(io.BytesIO(audio\_bytes))[1]                              self.labels = labels.to\_dict("records")[::subsample\_rate]         self.all\_audio = all\_audio         self.device = device      def \_\_len\_\_(self):         return len(self.labels)      def \_\_getitem\_\_(self, item):         record = self.labels[item]         audio = torch.from\_numpy(self.all\_audio[record["file\_name"]].copy())         text = record["transcription"]                  return (audio, text) |

|  |
| --- |
| dataset = Fleurs(lang, subsample\_rate=10)  # subsample 10% of the dataset for a quick demo |

**Running inference on the dataset using a medium Whisper model**

حالا بر روی یک مدل medium آموزش را ادامه می دهیم

|  |
| --- |
| model = whisper.load\_model("medium") print(     f"Model is {'multilingual' if model.is\_multilingual else 'English-only'} "     f"and has {sum(np.prod(p.shape) for p in model.parameters()):,} parameters." ) |

|  |
| --- |
| options = dict(language=language, beam\_size=5, best\_of=5) transcribe\_options = dict(task="transcribe", \*\*options) translate\_options = dict(task="translate", \*\*options) |

|  |
| --- |
| references = [] transcriptions = [] translations = []  for audio, text in tqdm(dataset):     transcription = model.transcribe(audio, \*\*transcribe\_options)["text"]     translation = model.transcribe(audio, \*\*translate\_options)["text"]          transcriptions.append(transcription)     translations.append(translation)     references.append(text) |

|  |
| --- |
| data = pd.DataFrame(dict(reference=references, transcription=transcriptions, translation=translations)) data |

با استفاده از کتابخانه ی tqdm برای داده های متنی و صوتی مدل transcribe ,translate را اعمال می

کنیم و دیتا های ترجمه و رونویسی شده را نمایش می دهیم

**خروجی در کولب**

**بقیه ی خروجی ها به علت زیاد بودن در پوشه ی output ضمیمه شده است**

****

**Word-level timestamps using attention weights**

در زیر، از weights cross-attention برای تعیین timestamps دقیق‌تر و در سطح کلمه استفاده می‌کنیم. از مجموعه ای از روش های heuristics و زمان پویا (DTW)برای یافتن تراز بین صدا و transcriptاستفاده می شود.

ابتدا کتابخانه های مهم ایمپورت و dtw رانصب می کنیم

|  |
| --- |
| ! pip install dtw-python |

|  |
| --- |
| import string import matplotlib.pyplot as plt import matplotlib.font\_manager as fm import matplotlib.ticker as ticker  from IPython.display import display, HTML from whisper.tokenizer import get\_tokenizer from dtw import dtw from scipy.ndimage import median\_filter  %matplotlib inline %config InlineBackend.figure\_format = "retina" |

مدل چند زبانه را در قطعه کد زیر tokenize می کنیم

|  |
| --- |
| AUDIO\_SAMPLES\_PER\_TOKEN = whisper.audio.HOP\_LENGTH \* 2 AUDIO\_TIME\_PER\_TOKEN = AUDIO\_SAMPLES\_PER\_TOKEN / whisper.audio.SAMPLE\_RATE  medfilt\_width = 7 qk\_scale = 1.0  tokenizer = get\_tokenizer(model.is\_multilingual, language=languages[lang]) |

زبان های چینی و کره ای و ژاپنی فونت مخصوص به خود GoNoteJKCore را دارند .سایر زبان ها مانند زبان فارسی فونت GoNotoCurrent دارند

|  |
| --- |
| # This part downloads a repackaged version of the Noto Sans font (either CJK or non-CJK) # to render various languages in Matplotlib figures.  if languages[lang] in {"Chinese", "Japanese", "Korean"}:     font = "GoNotoCJKCore.ttf" else:     font = "GoNotoCurrent.ttf"  font\_release = "https://github.com/satbyy/go-noto-universal/releases/download/v5.2" if not os.path.exists(font):     download(f"{font\_release}/{font}", font)  prop = fm.FontProperties(fname=font) props = {'fontproperties': prop} |

Unicode و space ها را tokenize می کنیم و سپس روی timespace ها در سطح یونیکد decode می کنیم و punctuation ها و کاراکتر های ویژه و فضاهای خالی را subword در نظر می گیریم

|  |
| --- |
| def split\_tokens\_on\_unicode(tokens: torch.Tensor):     words = []     word\_tokens = []     current\_tokens = []          for token in tokens.tolist():         current\_tokens.append(token)         decoded = tokenizer.decode\_with\_timestamps(current\_tokens)         if "\ufffd" not in decoded:             words.append(decoded)             word\_tokens.append(current\_tokens)             current\_tokens = []          return words, word\_tokens |

|  |
| --- |
| def split\_tokens\_on\_spaces(tokens: torch.Tensor):     subwords, subword\_tokens\_list = split\_tokens\_on\_unicode(tokens)     words = []     word\_tokens = []          for subword, subword\_tokens in zip(subwords, subword\_tokens\_list):         special = subword\_tokens[0] >= tokenizer.eot         with\_space = subword.startswith(" ")         punctuation = subword.strip() in string.punctuation         if special or with\_space or punctuation:             words.append(subword)             word\_tokens.append(subword\_tokens)         else:             words[-1] = words[-1] + subword             word\_tokens[-1].extend(subword\_tokens)          return words, word\_tokens |

اگر زبان ما یکی از پنج زبان چینی و ژاپنی و میانماری و .. بود بر اساس یونیکد ها tokenize می کنیم و در غیر این صورت بر اساس فاصله ها tokenize می کنیم

|  |
| --- |
| if languages[lang] in {"Chinese", "Japanese", "Thai", "Lao", "Myanmar"}:     # These languages don't typically use spaces, so it is difficult to split words     # without morpheme analysis. Here, we instead split words at any     # position where the tokens are decoded as valid unicode points     split\_tokens = split\_tokens\_on\_unicode else:     split\_tokens = split\_tokens\_on\_spaces |

وزن های attention را بر اساس cross-attention layers بدست می آوریم

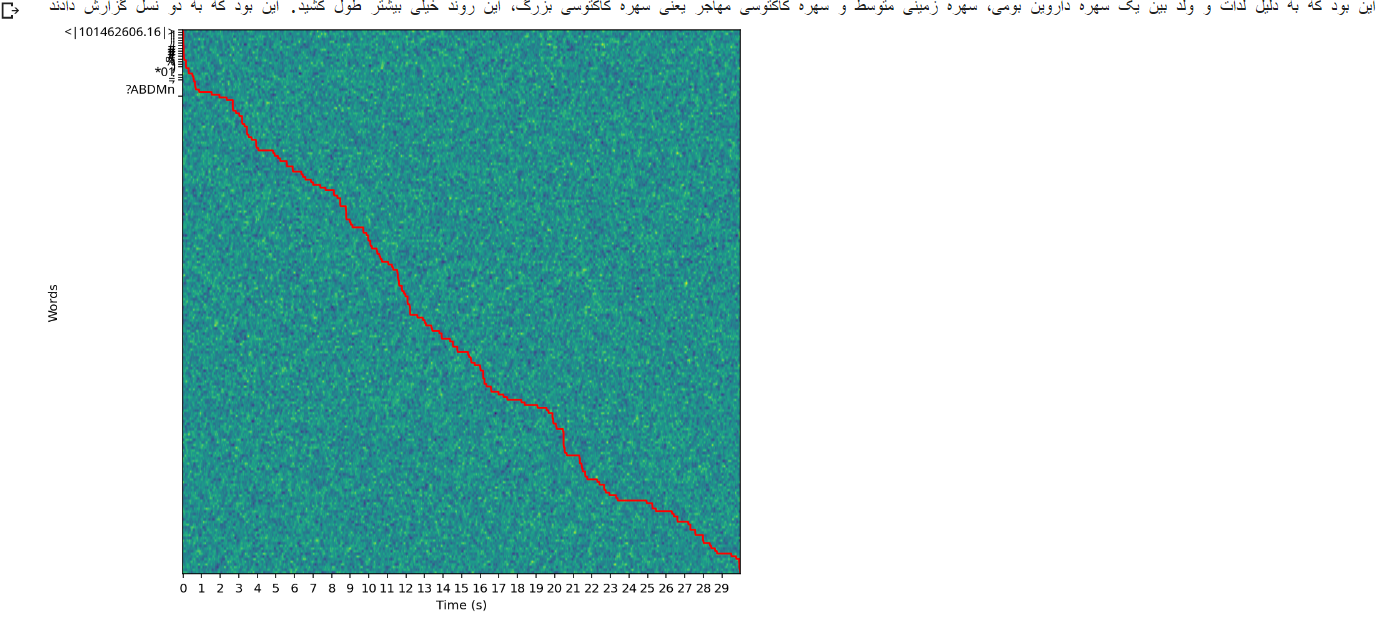
|  |
| --- |
| # install hooks on the cross attention layers to retrieve the attention weights QKs = [None] \* model.dims.n\_text\_layer  for i, block in enumerate(model.decoder.blocks):     block.cross\_attn.register\_forward\_hook(         lambda \_, ins, outs, index=i: QKs.\_\_setitem\_\_(index, outs[-1])     ) |

**رسم dtw برای ده مثال اول در دیتاست**

|  |
| --- |
| # for the first 10 examples in the dataset  for (audio, label), transcription in zip(dataset, transcriptions[:10]):     print(transcription)        duration = len(audio)     mel = whisper.log\_mel\_spectrogram(whisper.pad\_or\_trim(audio)).cuda()     tokens = torch.tensor(         [             \*tokenizer.sot\_sequence,             tokenizer.timestamp\_begin,         ] + tokenizer.encode(transcription) + [             tokenizer.timestamp\_begin + duration // AUDIO\_SAMPLES\_PER\_TOKEN,             tokenizer.eot,         ]     ).cuda()     with torch.no\_grad():         logits = model(mel.unsqueeze(0), tokens.unsqueeze(0))      weights = torch.cat(QKs)  # layers \* heads \* tokens \* frames         weights = weights[:, :, :, : duration // AUDIO\_SAMPLES\_PER\_TOKEN].cpu()     weights = median\_filter(weights, (1, 1, 1, medfilt\_width))     weights = torch.tensor(weights \* qk\_scale).softmax(dim=-1)          w = weights / weights.norm(dim=-2, keepdim=True)     matrix = w[-6:].mean(axis=(0, 1))      alignment = dtw(-matrix.double().numpy())      jumps = np.pad(np.diff(alignment.index1s), (1, 0), constant\_values=1).astype(bool)     jump\_times = alignment.index2s[jumps] \* AUDIO\_TIME\_PER\_TOKEN     words, word\_tokens = split\_tokens(tokens)      # display the normalized attention weights and the alignment     plt.figure(figsize=(8, 8))     plt.imshow(matrix, aspect="auto")     plt.plot(alignment.index2s, alignment.index1s, color="red")      xticks = np.arange(0, matrix.shape[1], 1 / AUDIO\_TIME\_PER\_TOKEN)     xticklabels = (xticks \* AUDIO\_TIME\_PER\_TOKEN).round().astype(np.int32)      plt.xticks(xticks, xticklabels)     plt.xlabel("Time (s)")          # display tokens and words as tick labels     ylims = plt.gca().get\_ylim()      ax = plt.gca()     ax.tick\_params('both', length=0, width=0, which='minor', pad=6)      ax.yaxis.set\_ticks\_position("left")     ax.yaxis.set\_label\_position("left")     ax.invert\_yaxis()     ax.set\_ylim(ylims)      major\_ticks = [-0.5]     minor\_ticks = []     current\_y = 0          for word, word\_token in zip(words, word\_tokens):         minor\_ticks.append(current\_y + len(word\_token) / 2 - 0.5)         current\_y += len(word\_token)         major\_ticks.append(current\_y - 0.5)              ax.yaxis.set\_minor\_locator(ticker.FixedLocator(minor\_ticks))     ax.yaxis.set\_minor\_formatter(ticker.FixedFormatter(words))     ax.set\_yticks(major\_ticks)     ax.yaxis.set\_major\_formatter(ticker.NullFormatter())          for label in ax.get\_yminorticklabels():         label.set\_fontproperties(prop)      plt.ylabel("Words")     plt.show()      # display the word-level timestamps in a table     word\_boundaries = np.pad(np.cumsum([len(t) for t in word\_tokens[:-1]]), (1, 0))     begin\_times = jump\_times[word\_boundaries[:-1]]     end\_times = jump\_times[word\_boundaries[1:]]      data = [         dict(word=word, begin=begin, end=end)         for word, begin, end in zip(words[:-1], begin\_times, end\_times)         if not word.startswith("<|") and word.strip() not in ".,!?、。"     ]      display(pd.DataFrame(data))     display(HTML("<hr>")) |

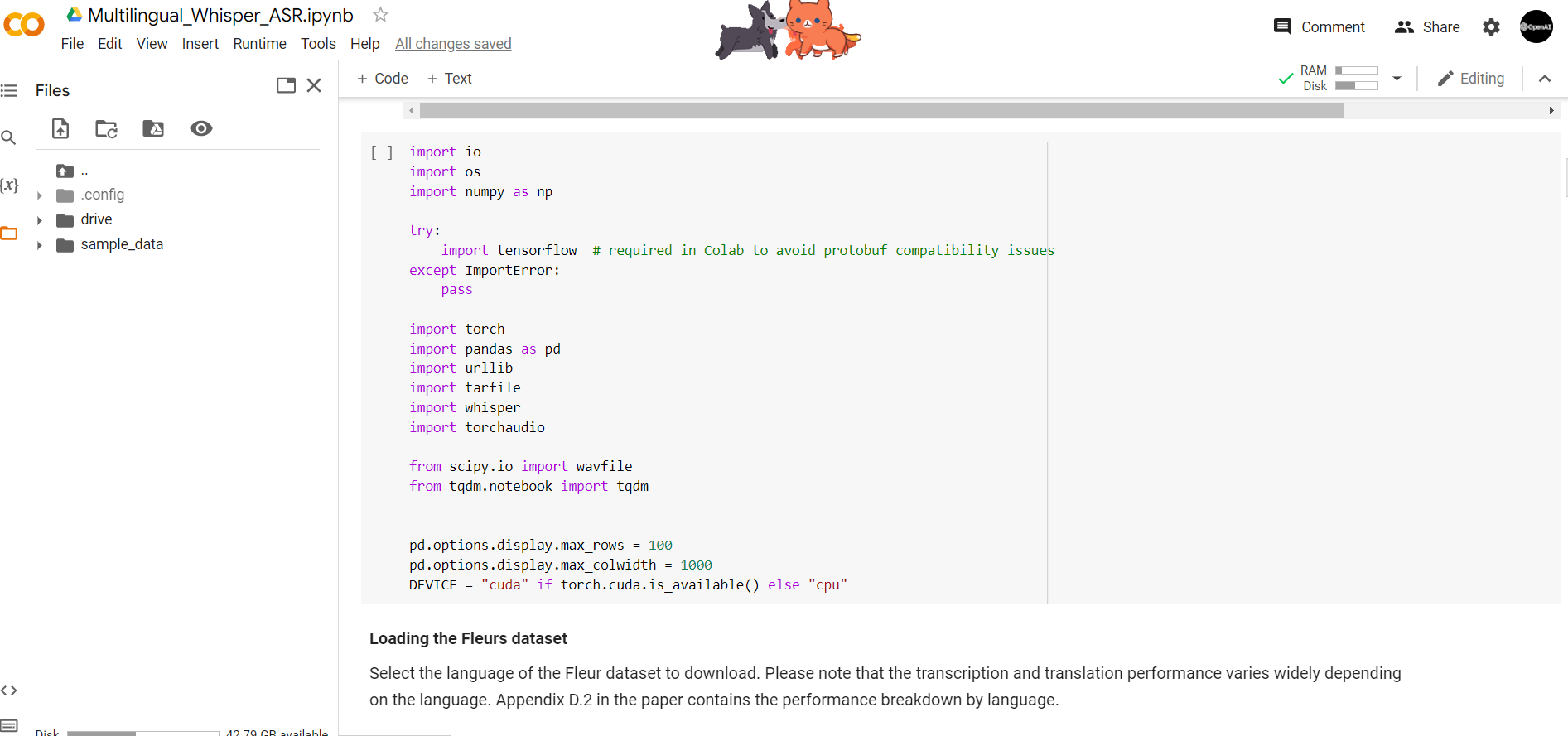
ابتدا log\_mel\_spectrogram را محاسبه می کنیم . سپس بر اساس timestamps توکنایز کرده و وزن ها را بدست آورده و نرمالیز می کنیم سپس الگوریتم dtw را برای بدست آوردن کمترین فاصله ی بین عبارات اعمال می کنیم و با matplotlib نمودار را رسم می کنیم

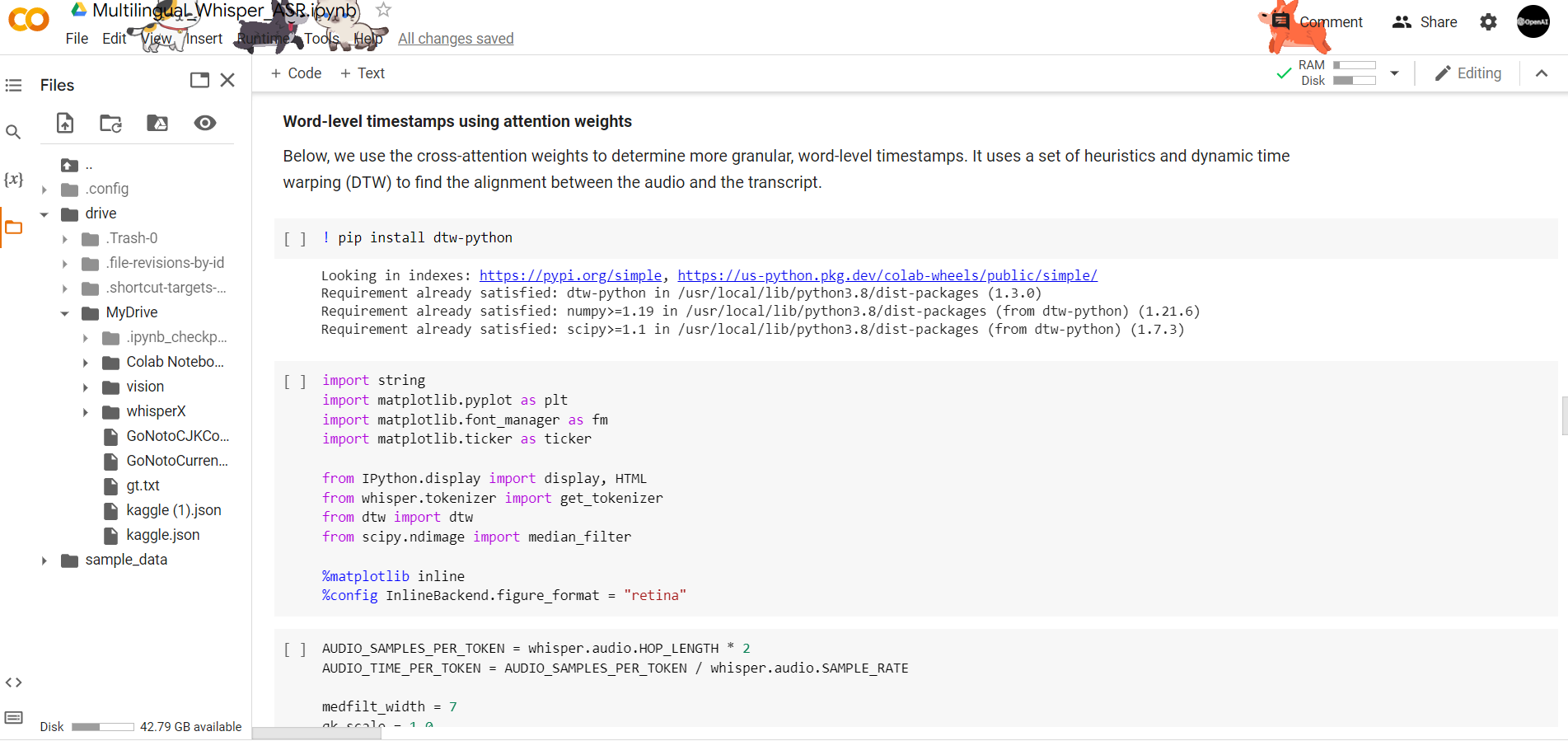
**خروجی**

****همانطور که در خروجی زیر مشخص است با توجه به اعمال dtw در دیتاست فلور کلماتی که فاصله ی کمتری به هم دارند یعنی به هم نزدیک تر اند پیدا شده اند مثلا سهره زمینی متوسط و سهره کاکتوسی مهاجر یعنی سهره کاکتوسی بزرگ

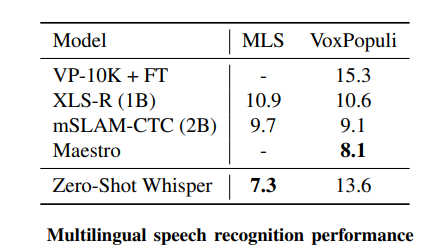
|  |
| --- |
| محاققان دانشگاه پرینستون امریکا و دانشگاه اوبسلا در سوید گنهای جدید تکمل یافتهی را تنها در دو نسل گزارش دادند. اگرچه اعتقاد بر این بود که به دلیل لذات و ولد بین یک سهره داروین بومی، سهره زمینی متوسط و سهره کاکتوسی مهاجر یعنی سهره کاکتوسی بزرگ، این روند خیلی بیشتر طول کشید. این بود که به دو نسل گزارش دادند. |

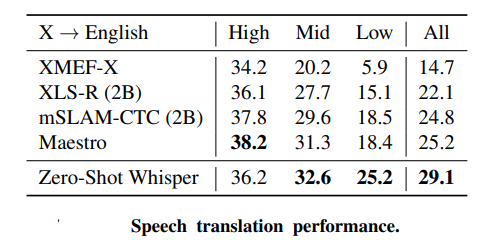
**خروجی در محیط کولب**

****



**مدل چند زبانه و ارزیابی مدل**

در جدول Whisper در LibriSpeech چند زبانه عملکرد خوبی دارد و از XLS-R (Bapna) بهتر عمل می کند Zeroshot Whisper عملکرد را در LibriSpeech چند زبانه (MLS) بهبود می بخشد، اما همچنان به طور قابل توجهی از Maestro، XLS-R و mSLAM در VoxPopuli عقب است**.**

در شکل زیر همبستگی بین مقدار داده های آموزش ترجمه در هر زبان و امتیاز BLEU zero shot حاصل در Fleurs را رسم می کنیم. در حالی که روند بهبود با افزایش داده های آموزشی وجود دارد، ضریب همبستگی مجذور بسیار کمتر از 0.83 مشاهده شده برای تشخیص گفتار و تنها 0.24 است. فرض می کنیم که این تا حدی به دلیل داده های آموزشی نویزی به دلیل خطا در شناسایی زبان صوتی باشد.

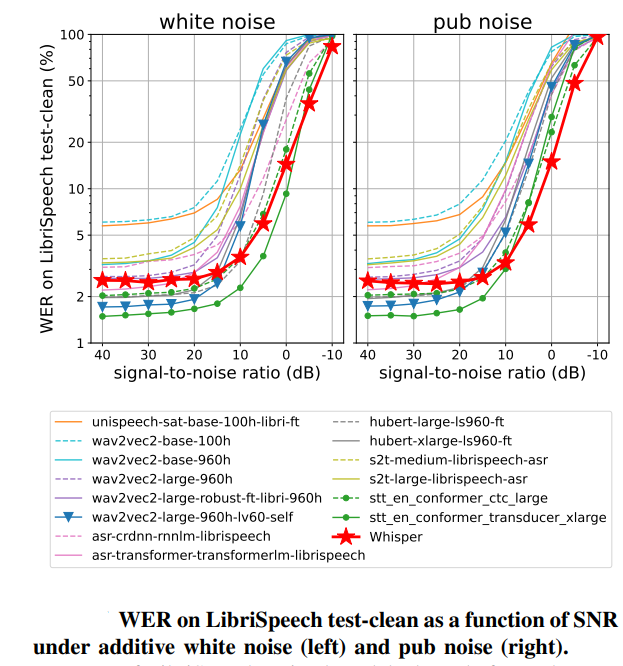
**ارزیابی Whisper در مجموعه داده LibriSpeech ASR.**

|  |
| --- |
| pip   install transformers datasets evaluate |

|  |
| --- |
| from datasets import load\_dataset from transformers import WhisperForConditionalGeneration, WhisperProcessor import torch from evaluate import load |
| librispeech\_test\_clean = load\_dataset("librispeech\_asr", "clean", split="test")  processor = WhisperProcessor.from\_pretrained("openai/whisper-base.en") model = WhisperForConditionalGeneration.from\_pretrained("openai/whisper-base.en").to("cuda")  def map\_to\_pred(batch):     audio = batch["audio"]     input\_features = processor(audio["array"], sampling\_rate=audio["sampling\_rate"], return\_tensors="pt").input\_features     batch["reference"] = processor.tokenizer.\_normalize(batch['text'])      with torch.no\_grad():         predicted\_ids = model.generate(input\_features.to("cuda"))[0]     transcription = processor.decode(predicted\_ids)     batch["prediction"] = processor.tokenizer.\_normalize(transcription)     return batch  result = librispeech\_test\_clean.map(map\_to\_pred)  wer = load("wer") print(100 \* wer.compute(references=result["reference"], predictions=result["prediction"])) |

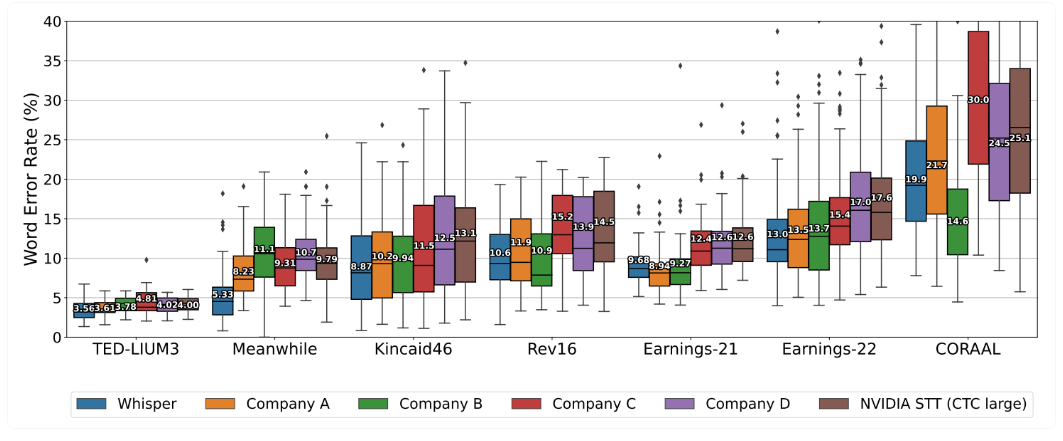
**خروجی**

|  |
| --- |
| 4.254436419182681 |

****

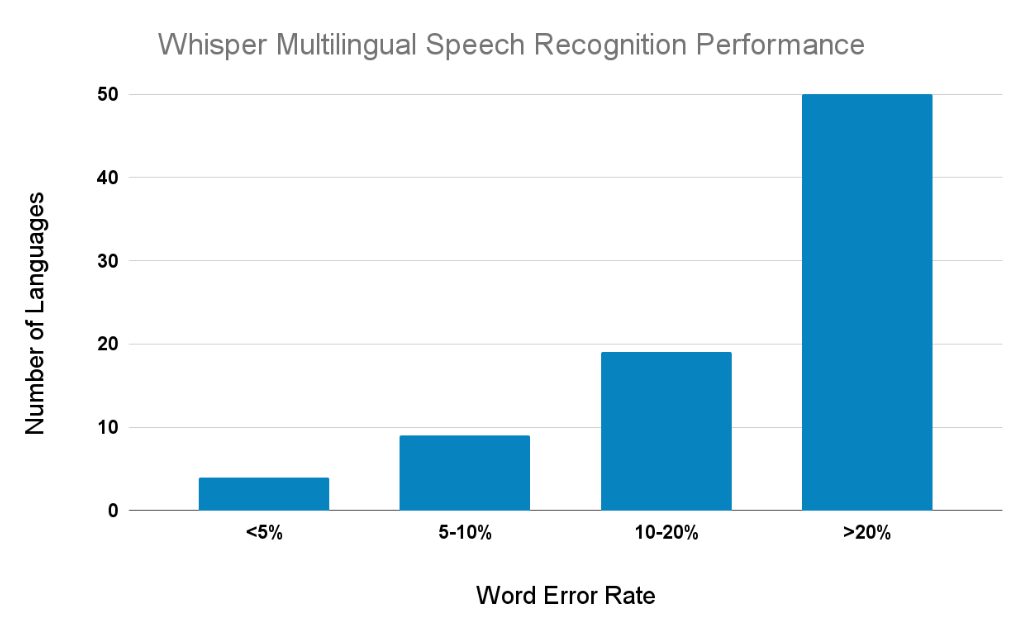
WER در LibriSpeech test-clean به عنوان تابعی از SNR تحت نویز سفید افزودنی (سمت چپ) و نویز pub(راست). دقت مدل های آموزش دیده با LibriSpeech سریعتر از بهترین مدل Whispermodel (F) کاهش می یابد. مدل‌های NVIDIA STT (•) در شرایط نویز کم بهترین عملکرد را دارند، اما با Whisper در نویز زیاد (SNR < 10 دسی‌بل) بهتر عمل می‌کنند. دومین مدل برتر تحت نویز کم H فقط در LibriSpeech تنظیم می شود و حتی سریع تر از بین می رود.

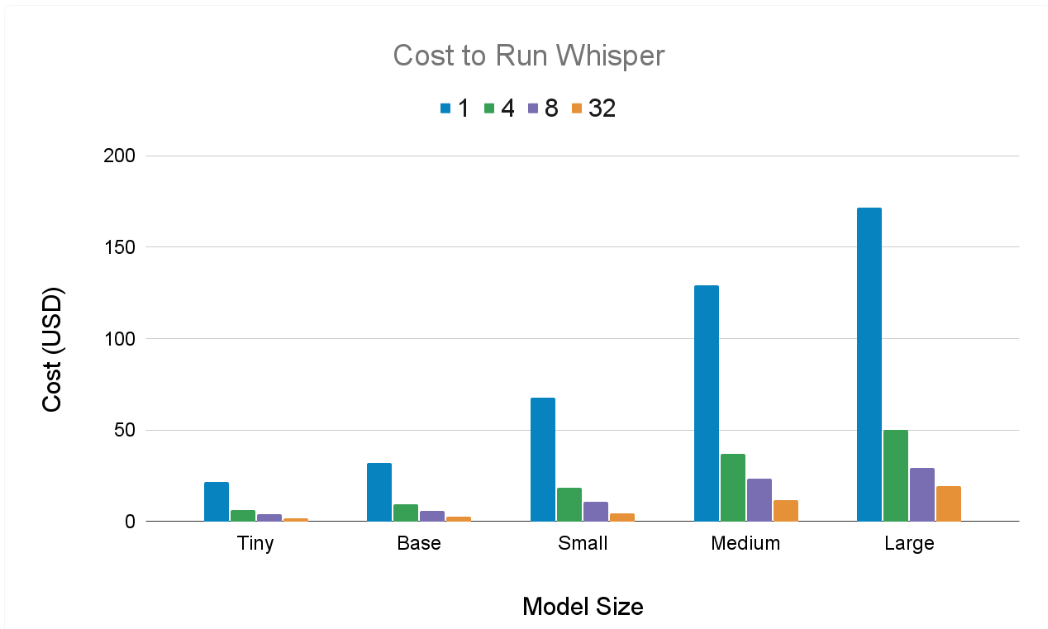
**OpenAI Whisper Analysis**

شکل زیر دقت Whisper را با استفاده از Word-Error-Rate (WER) با مدل‌های تشخیص گفتار پیشرفته فعلی مقایسه می‌کند. به خصوص با توجه به اینکه Whisper یک مدل منبع باز است.

در حالی که این نتایج عالی هستند، تشخیص گفتار همچنان یک open problem به خصوص برای زبان های غیر انگلیسی است. شکل زیر میزان خطای Word Whisper را برای هر زبان پشتیبانی شده گزارش می‌کند. در حالی که Whisper در چندین زبان مانند، آلمانی، ژاپنی و غیره به نتایج پیشرفته‌ای دست می‌یابد، عملکرد نسبتاً برای سایر زبان‌ها کم است.

در زیر توزیع زبان ها را تابعی از میزان خطای کلمه می بینیم. از 82 زبان موجود ، 50 مورد از آنها دارای نرخ خطای کلمه بیش از 20٪ هستند.

**هزینه اجرای Whisper**

 در نمودار زیر هزینه transcribe 1000 ساعت صدا را با استفاده از Whisper در GCP (1x A100 40 گیگابایت) برای هر اندازه مدل با استفاده از اندازه‌های دسته‌ای مختلف ارائه می شود‌

**Final Words**

تجزیه و تحلیل فوق نشان داد که Whisper به نتایج پیشرفته‌ای برای تشخیص گفتار در چندین زبان دست می‌یابد. عملکرد Whisper تا حدی از شدت محاسبات آن ناشی می‌شود، بنابراین برنامه‌هایی که به نسخه‌های بزرگتر و قدرتمندتر Whisper نیاز دارند، باید مطمئن شوند که Whisper را روی GPU به صورت محلی یا در فضای ابری اجرا می‌کنند