

Parakeet frente a Whisper Turbo y Distil en CPU

Asad Shehzad

Loudness.ai · LA-RB-2026-001 · 14 October 2026 · Version 1.0
DOI: [10.5281/zenodo.23374001](https://doi.org/10.5281/zenodo.23374001)

Resumen

Evaluamos tres configuraciones de Parakeet TDT 0.6B v3 y dos variantes cuantizadas de Whisper con el mismo audio de lectura en inglés y cuatro hilos de CPU. En una grabación de 310,425 segundos, Parakeet FP32 con segmentos cortos tardó 53,58 segundos, con una tasa de error por palabra del 2,81% (23 ediciones sobre 818 palabras de referencia) y 2,87 GiB de memoria máxima del proceso. Whisper Turbo INT8 tardó 343,26 segundos con un WER del 2,20%; Distil-Whisper INT8 tardó 279,10 segundos con un WER del 2,57%. Los segmentos predeterminados de Parakeet INT8 omitieron una cantidad considerable de voz; los segmentos cortos redujeron el WER del 39,12% al 4,65%. Estas observaciones por solicitud favorecen a Parakeet FP32 en esta CPU y muestra, aunque las variantes de Whisper evaluadas cometieron menos errores. Publicamos las 18 ejecuciones completadas, transcripciones, audios de prueba, referencias, sumas de verificación y ajustes de reproducción.

Cinco configuraciones, una grabación

Cada fila utiliza el mismo WAV de 310,425 segundos y 818 palabras, verificado mediante SHA-256. Los experimentos usaron una cuota equivalente a cuatro núcleos y cuatro hilos de inferencia en el mismo servidor virtual compartido. No se evaluó el modelo completo Whisper large-v3; la comparación incluye únicamente las variantes Turbo y Distil.

Whisper Turbo tardó 6,41 veces más que Parakeet FP32; Distil tardó 5,21 veces más. Sus recuentos de errores fueron menores: 18 y 21 ediciones, frente a 23 de Parakeet FP32. Parakeet INT8 con segmentos cortos usó menos memoria, pero sus 38 ediciones superaron las 23 de FP32. Son solicitudes completadas, no estimaciones de rendimiento.

Configuración	Time (s)	WER	Edits	RSS (GiB)
Whisper Turbo INT8	343.26	2.20%	18/818	2.09
Distil-Whisper INT8	279.10	2.57%	21/818	1.96
Parakeet INT8 · default	68.22	39.12%	320/818	2.02
Parakeet INT8 · short	53.01	4.65%	38/818	1.46
Parakeet FP32 · short	53.58	2.81%	23/818	2.87

Todas las ejecuciones completadas

El estudio contiene doce ejecuciones de Parakeet y seis de Whisper. Cada configuración larga, limpia o con ruido, se midió una sola vez. Solo las grabaciones limpias cortas incluyen una primera solicitud y una repetición con el modelo cargado. Este diseño no permite barras de error ni intervalos de confianza. La descarga incluye métricas con precisión completa, tiempo de CPU, memoria muestreada, promedios de carga y transcripciones.

Los nombres 30sec y 5min son etiquetas: las duraciones exactas son 36,85 y 310,425 segundos. Se conservó una muestra de 1.203,660 segundos, pero su ejecución anterior de Whisper se interrumpió sin resultado completo y queda excluida de las tablas.

Configuration / case	Time (s)	Edits / ref.	WER	RSS (GiB)
INT8, default chunks / 30sec-first-request	6.11	25/93	26.88%	1.29
INT8, default chunks / 30sec-warm-repeat	5.89	25/93	26.88%	1.52
INT8, default chunks / 5min-warm	68.22	320/818	39.12%	2.02
INT8, default chunks / 5min-noise10db-warm	53.73	111/818	13.57%	2.18
INT8, short chunks / 30sec-first-request	8.27	9/93	9.68%	1.35
INT8, short chunks / 30sec-warm-repeat	5.81	9/93	9.68%	1.40
INT8, short chunks / 5min-warm	53.01	38/818	4.65%	1.46
INT8, short chunks / 5min-noise10db-warm	57.27	64/818	7.82%	1.46
FP32, short chunks / 30sec-first-request	9.51	3/93	3.23%	2.79
FP32, short chunks / 30sec-warm-repeat	6.05	3/93	3.23%	2.81
FP32, short chunks / 5min-warm	53.58	23/818	2.81%	2.87
FP32, short chunks / 5min-noise10db-warm	59.20	37/818	4.52%	2.87
Whisper Turbo INT8 / 30sec-first-request	70.48	2/93	2.15%	1.97
Whisper Turbo INT8 / 30sec-warm-repeat	66.04	2/93	2.15%	1.97
Whisper Turbo INT8 / 5min-warm	343.26	18/818	2.20%	2.09
Distil-Whisper INT8 / 30sec-first-request	49.85	2/93	2.15%	1.83
Distil-Whisper INT8 / 30sec-warm-repeat	42.57	2/93	2.15%	1.84
Distil-Whisper INT8 / 5min-warm	279.10	21/818	2.57%	1.96

Hardware y límites de medición

La máquina virtual KVM exponía 12 CPU virtuales AMD EPYC, x86-64 con AVX2 y unos 47 GiB de RAM. No exponía VNNI. Esto no determina la frecuencia de los núcleos físicos ni el rendimiento de una máquina dedicada. La inferencia tenía una cuota equivalente a cuatro núcleos, cuatro hilos, prioridad de planificación baja y ningún uso permitido de memoria de intercambio. La instalación y descarga se ejecutaron por separado; la inferencia completada utilizó archivos locales sin conexión externa.

El tiempo de Parakeet incluye la carga HTTP local, el análisis multipart, la decodificación WAV, la detección de actividad de voz, la segmentación, la inferencia, la unión de resultados, las marcas de tiempo, la

serialización JSON y la recepción. Whisper mide la llamada directa al proceso OpenVINO, incluida la conversión de entrada. Ambos excluyen el inicio del modelo, la lectura del disco, la puntuación, la carga por Internet y la espera en cola. Los límites diferentes corresponden a una comparación de aplicaciones, no a una comparación de operaciones con el mismo entorno.

La memoria RSS de Parakeet se muestreó cada 50 milisegundos e incluye las páginas del modelo y las bibliotecas. Excluye el cliente y difiere del cálculo agregado del sistema operativo. La tabla de cinco configuraciones usa el máximo muestreado de cada solicitud. Las pruebas fueron secuenciales, en momentos diferentes, sobre un servidor compartido; se observó la carga de fondo sin controlarla. No se midió la capacidad con usuarios simultáneos.

Modelos y ajustes exactos

Parakeet usó la [implementación FastAPI de groxaxo](#) en el commit `06464cd06a0f75426ce46e94c2d04ee7e26f48c0`, con la [exportación ONNX de istupakov](#) en la revisión `8f23f0c03c8761650bdb5b40aaf3e40d2c15f1ce`. INT8 y FP32 cargaron archivos de precisión distinta de esa revisión. El modelo y el código de decodificación no cambiaron; los ajustes se aplicaron mediante variables de entorno. Se verificó la ejecución exclusivamente en CPU del codificador y del decodificador conjunto.

Entorno Parakeet: Python 3.12.3, ONNX Runtime CPU 1.23.2, onnx-asr 0.12.0, NumPy 1.26.4, PyTorch 2.8.0+cpu, SoundFile 0.13.1 y Silero VAD 6.2.3. Ajustes: cuatro hilos internos, un hilo entre operaciones, ejecución secuencial, optimización completa del grafo, un trabajador de inferencia, uno de audio, uno web y agrupación entre solicitudes desactivada. Los grupos BLAS y OpenMP tenían un hilo.

Los segmentos predeterminados usaron duraciones mínima/objetivo/máxima de 20/60/75 segundos. Ambos perfiles cortos usaron 5/12/15 segundos. La detección de voz y los límites de pausa forman parte del comportamiento evaluado. La grabación larga limpia produjo 30 segmentos con INT8 y 31 con FP32. Esa diferencia impide atribuir todos los cambios de calidad exclusivamente a la precisión aritmética. El inicio de cada perfil de Parakeet fue de aproximadamente 8,8 segundos; el primer uso de la detección de voz afectó a las solicitudes cortas iniciales.

Whisper usó [OpenVINO Whisper large-v3 Turbo INT8](#) en la revisión `b568445dd5dc8c695bde596f8acbb4694fd6ba64` y [OpenVINO Distil-Whisper large-v3 INT8](#) en la revisión `7210c37ac7fb59669a9768fe194c5fe022ddc743`. Entorno: OpenVINO 2026.1.0, OpenVINO GenAI 2026.1.0.0 y Python 3.12.3; cuatro hilos, un flujo, modo de latencia, decodificación voraz con inglés especificado y marcas por segmento. El inicio fue de 4,88 segundos para Turbo y 3,90 para Distil, medido por separado.

Procedencia del audio y puntuación

La fuente es [LibriSpeech test-clean \(OpenSLR 12\)](#), de Vassil Panayotov y colaboradores, derivada de audiolibros de LibriVox y distribuida bajo [CC BY 4.0](#). El archivo pasó la comprobación MD5 publicada `32fa31d27d2e1cad72775fee3f4849a9` y SHA-256 `39fde525e59672dc6d1551919b1478f724438a95aa55f874b576be21967e6c23`. Las muestras usan ordenación aleatoria con semilla 42, enunciados completos, pausas de 0,25 segundos y WAV PCM16 mono de 16 kHz. El paquete incluye identificadores, selección de hablantes, referencias y hashes exactos.

La muestra corta contiene 93 palabras de referencia en cuatro enunciados; la larga contiene 818 palabras. Los identificadores cortos son 4970-29093-0015, 5639-40744-0030, 8555-292519-0011 y 121-127105-0034. Los archivos limpios de Parakeet son idénticos byte a byte a los de Whisper. Son lecturas concatenadas de varios hablantes, no reuniones continuas.

La muestra con ruido añade ruido blanco gaussiano con semilla 42 a la grabación larga, con una relación señal-ruido de energía global de 10 dB. Si el pico supera 0,98, se reduce uniformemente el nivel antes de codificar en PCM16. La relación registrada antes de codificar fue de 9,9999996 dB y la ganancia de 1,0. Es ruido estacionario sintético, no tráfico ni ruido real de sala. Whisper no se evaluó con este archivo.

El WER es la distancia exacta de edición de palabras de Levenshtein dividida por el número de palabras de referencia. Se convierten las letras a minúsculas y los apóstrofes curvos a rectos; la tokenización conserva palabras alfanuméricas y apóstrofes internos. Se incluye la expresión regular y el código de puntuación. El WER mide sustituciones, omisiones e inserciones; no evalúa directamente puntuación, mayúsculas, identificación del hablante, nombres importantes ni legibilidad de subtítulos.

Omisiones, segmentación y ruido

La transcripción corta de INT8 predeterminado tenía 69 palabras normalizadas frente a 93 de referencia y omitió los dos enunciados finales. La larga tenía 513 frente a 818. En la solicitud corta, el último fragmento de voz real terminó cerca de los 25,5 segundos, mientras que el sistema prolongó la puntuación final y el segmento hasta los 36,85 segundos del audio. Por tanto, una marca final igual a la duración no demuestra reconocimiento completo.

El campo words contiene fragmentos del tokenizador como testi, mon e ies, en lugar de palabras completas reagrupadas. Los finales de segmento pueden derivarse de los límites de los fragmentos. Se conservan esas respuestas; la alineación fiable por palabra y la legibilidad de subtítulos necesitan una evaluación independiente.

Los segmentos cortos redujeron el WER de INT8 en la grabación larga limpia del 39,12% al 4,65%. FP32 lo redujo al 2,81%, con un tiempo casi idéntico. Con ruido, el WER fue del 13,57% para INT8 predeterminado, 7,82% para INT8 corto y 4,52% para FP32 corto. La mejora de INT8 predeterminado con ruido es inusual y podría implicar la detección de voz o la decodificación; no significa que el ruido mejore en general el reconocimiento.

La [documentación de cuantización de ONNX Runtime](#) describe una posible saturación U8S8 en hardware AVX2. Es una hipótesis plausible para parte del deterioro INT8, no una causa demostrada. No probamos cuantización de rango reducido, otras exportaciones ni límites de segmentos idénticos controlados.

Qué permiten concluir estos resultados

Para esta CPU virtual, exportación y pequeña muestra en inglés, Parakeet FP32 con segmentos cortos ofrece el mejor equilibrio observado entre velocidad y errores de los perfiles Parakeet evaluados. Funcionó sin conexión después de la instalación. Es una elección de configuración respaldada por solicitudes medidas, no una afirmación de que Parakeet sea universalmente más preciso que Whisper.

Según la [ficha de NVIDIA](#), el modelo Parakeet admite inglés y otros 24 idiomas europeos; este experimento evalúa solo inglés. No permite conclusiones para urdu, hindi, árabe, alternancia de idiomas ni acentos concretos. Distil es solo para inglés; Turbo es multilingüe, pero no se midió ese comportamiento.

Las amenazas a la validez incluyen la muestra pequeña concatenada, cambios abruptos de hablante y contexto, una sola medición por archivo largo, carga compartida sin controlar, límites de medición diferentes y límites de voz no idénticos. No medimos precisión sobre un corpus completo, latencia extrema, concurrencia, diferencias entre CPU físicas, ruido real ni el modelo completo Whisper large-v3. Un estudio más amplio debe añadir corpus independientes, repeticiones, controles de segmentación, pruebas de

conurrencia y evaluación de marcas de tiempo.

Reproducir e inspeccionar el estudio

El paquete incluye las 18 ejecuciones, comparaciones CSV/JSON, doce respuestas de Parakeet, seis respuestas por segmentos de Whisper, transcripciones, muestras WAV limpias y con ruido, referencias, selección original de enunciados, hashes del modelo, versiones de dependencias y scripts portátiles. Los scripts usan la implementación original fijada y admiten los tres perfiles evaluados; reproducen los límites de solicitud y puntuación. Los modelos se descargan por separado con sus licencias originales. El investigador debe aplicar los límites de CPU y el aislamiento de red.

Prepare entornos Python aislados con las versiones incluidas, obtenga las revisiones de código y modelos, verifique los hashes y use los WAV suministrados. Configure cuatro hilos y una cuota equivalente a cuatro núcleos. Ejecute cada perfil de Parakeet por separado con un trabajador y modelos locales; después, los dos procesos de Whisper con inglés especificado. Separe el inicio de la medición, muestree RSS de Parakeet cada 50 milisegundos y de Whisper cada 100 milisegundos y puntúe con las referencias. Conserve las respuestas y comprobaciones del entorno antes de comparar resúmenes.

La ficha de Parakeet indica CC BY 4.0; las fichas de Whisper evaluadas indican MIT. El paquete no redistribuye pesos. Los audios y textos derivados de LibriSpeech conservan su atribución CC BY 4.0; las modificaciones son concatenación y ruido sintético. El informe técnico firmado lo publica Loudness.ai en su versión 1.0. El archivo original completo de evidencias internas se conserva separado del paquete público.

Referencias

Panayotov, V., Chen, G., Povey, D. y Khudanpur, S. (2015). LibriSpeech: an ASR corpus based on public domain audio books. *ICASSP 2015*, DOI: [10.1109/ICASSP.2015.7178964](https://doi.org/10.1109/ICASSP.2015.7178964). Corpus: [OpenSLR 12](#).

NVIDIA. [Ficha Parakeet TDT 0.6B v3](#). istupakov. [Exportación ONNX de Parakeet](#). groxaxo. [Implementación FastAPI de Parakeet](#). Las revisiones fijadas figuran en los métodos y el paquete.

OpenVINO. [Ficha Whisper large-v3 Turbo INT8](#); [Ficha Distil-Whisper large-v3 INT8](#). ONNX Runtime. [Cuantización de modelos ONNX: rendimiento y precisión](#). Fuentes consultadas el 14 de octubre de 2026.

Cita

Shehzad, Asad (2026). Parakeet vs Whisper Turbo and Distil on CPU. Loudness.ai technical report LA-RB-2026-001, version 1.0. <https://loudness.ai/research/parakeet-whisper-cpu-benchmark/>

DOI: [10.5281/zenodo.23374001](https://doi.org/10.5281/zenodo.23374001)

© 2026 Asad Shehzad / Loudness.ai