Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# An Intelligent Multimodal File Management System for Motor-Impaired Users Using Context-Aware Semantic Retrieval

**Gaurang Pandey**
Vellore Institute of Technology (VIT)
**Kirti Adhikari**
Vellore Institute of Technology (VIT)
**V Rajalakshmi**
v.rajalakshmi@vit.ac.in

Vellore Institute of Technology (VIT)

**Research Article**

**Keywords:**

**Posted Date:** November 25th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11736203/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# An Intelligent Multimodal File Management System for Motor-Impaired Users Using Context-Aware Semantic Retrieval

**Gaurang Pandey[1], Kirti Adhikari[1], and V. Rajalakshmi[1]***

*[1] School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai, Tamil Nadu 600127, India. *Corresponding author: v.rajalakshmi@vit.ac.in*

**Abstract: Users with motor impairments—such as cerebral palsy, muscular dystrophy, or spinal cord injury—experience severe fatigue with conventional keyboard-and-mouse interfaces. This paper presents an offline-first, multimodal file management system for motor accessibility that integrates local speech recognition (Whisper Base, 797 ms latency), webcam dwell-gaze tracking, single-switch input, context-aware hybrid retrieval, and multi-tier safety guardrails. A fine-tuned DistilRoBERTa classifier separates twelve filesystem intents with 100.00% (216/216) test accuracy and 99.86% ± 0.19% five-fold cross-validation accuracy (TF-IDF: 99.54%; regex: 93.52%), and spoken commands reach 93.06% downstream intent accuracy on Whisper Base (97.22% on Whisper Small). Slot extraction achieves a 97.02% micro-F1. On 240 held-out recordings covering four accents and noise down to 5 dB SNR, intent accuracy is 84.58% (Base) and 92.08% (Small). A five-stage hybrid retrieval pipeline attains a Mean Reciprocal Rank of 0.893 and Recall@5 of 96.67% on 150 queries, and the multi-tier confirmation mechanism intercepted 100% of destructive operations (30/30). In an ethics-approved pilot with eight motor-impaired participants, the system cut mean task time by 68.63% (78.41 s to 24.60 s; $t(7) = 14.98$, $p < 0.001$), reduced NASA-TLX workload by 61.68% and reached a System Usability Scale score of 86.88 (Grade A).**

**Keywords: Multimodal *user interfaces • Assistive computing • Voice user interfaces • Semantic file retrieval • Motor impairment***

## 1 Introduction

File management, which covers navigating directories, finding documents and organizing folders, is a basic part of everyday computing, yet desktop interfaces still follow the WIMP (Windows, Icons, Menus, Pointer) paradigm. That model assumes fine motor control, coordination between both hands and the stamina to operate a mouse and keyboard precisely. For people with cerebral palsy, cervical spinal cord injury, amyotrophic lateral sclerosis (ALS), Parkinsonian tremor, multiple sclerosis or muscular dystrophy, these devices cause pain, spasticity, rapid fatigue and high error rates [1, 2, 3]. While voice commands are the key hands-free option [1, 4, 5], commercially available software like Windows Voice Access and Dragon NaturallySpeaking treats voice as equivalent to a computer keyboard or mouse pointer, requiring users to either spell out directory names or navigate through numbered menus. Commercial systems also use cloud-based automatic speech recognition technology, leading to privacy issues for confidential local files, while using only voice poses the problem of ambiguity when names are similar. Therefore, the design of an interface should incorporate speech as well as webcam gaze tracking in order to specify the targets.

There are two additional issues that persist. The first issue here is that most people do not name files precisely; for example, they might say, "Please find the Excel file related to the budget meeting of last week." On the other hand, Windows Search and Spotlight mostly rely on lexical matching, so the use of semantic embeddings and fuzzy matching, as well as temporal filtering, is required [9, 10, 11, 12]. The second issue is that the voice

command to delete, move or rename a file causes a system state change, while bot computation errors lead to difficult recovery cases. Despite that, traditional voice assistants do not have a verification layer based on confidence calibration.

This article presents the design, implementation and evaluation of a multimodal, voice-driven file management system (VDFM) for people with upper-limb motor impairments. We do not claim novelty for the underlying open-source components (OpenAI Whisper, Sentence-Transformers and MediaPipe); our contribution lies in how they are integrated and evaluated (Fig. 1 summarizes the workflow).

### 1.1 Primary Research Contributions

- Unified Multimodal Architecture: speech, MediaPipe dwell-gaze and debounced single-switch input in one offline-first pipeline provided to different levels of motor capability.
- Offline Edge ASR and Robust Intent Understanding: Whisper Base with a fine-tuned DistilRoBERTa classifier (100.00% test accuracy; 93.06% on live speech), with no outbound telemetry observed from the evaluated pipelines.
- Command-Weighted WER (CW-WER): a task-oriented metric that correlates significantly more strongly with downstream command success than normalized WER (Williams' $t(69) = 9.07$, $p < 0.0001$).
- Five-Stage Hybrid Retrieval and Multi-Tier Safety: exact, token, fuzzy and dense matching with temporal filtering (MRR = 0.893), an open JSON command schema, and confirmation that intercepted 30/30 destructive operations.
- Pilot Usability Study: an ethics-approved study with eight motor-impaired participants and four controls against Keyboard/Mouse and Windows Voice Access baselines.

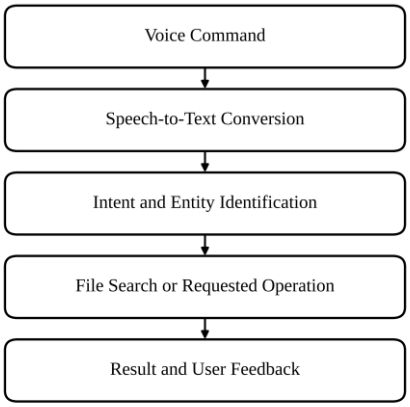


**Fig. 1** Functional workflow of the proposed voice-driven file management system.

## 2 Literature Review

Research on alternatives to physical input devices [1, 2, 3] shows that speech is particularly useful for people with motor disabilities. Harada et al. [1] showed with VoiceDraw that non-contact vocal interaction supports continuous tasks for users with severe impairments, and Saritha et al. [13] built a speech-driven code editor for programmers with upper-limb disabilities. Khan et al. [2] found in a scoping review that gaze for spatial selection combined with speech for semantic commands addresses both the 'Midas touch' problem of gaze and the deictic ambiguity of voice, and Bhattacharya et al. [3] modeled user errors in virtual scanning keyboards. Although

webcam gaze tracking [8] and single-switch devices provide dependable low-effort interfaces, previous research analyzed them in isolation, mostly in regard to communication boards or typing, and seldom in conjunction with natural language speech for hierarchical file system control. While OSWorld [7] benchmarks multimodal computer agents, such agents may use large models and do not suit users with motor impairments.

ASR has advanced from HMM and n-gram models to end-to-end neural systems. Yerramreddy and co-authors discovered that Whisper, which has been trained on 680,000 hours of weakly supervised data, exhibits strength to external noise and different accents. Atypical speech continues to be an issue: TORGO studies dysarthric pronunciation, and generic ASR systems fail when dealing with disordered speech, unless adapted appropriately. Converting transcripts to input commands requires determining intent and filling slots. Early file assistants used regular expressions or keyword lists, but both possess weaknesses in regard to paraphrasing and ASR errors; TF-IDF used with linear techniques is efficient on edge computers, while pre-trained transformers such as BERT, RoBERTa, and DistilRoBERTa perform better on colloquial language.

The conventional approach to file search involves using inverted indexes and lexical measures like BM25 [9, 10, 12, 28], which are ineffective for partial or descriptive queries. Dense bi-encoder methods, including Sentence-BERT [10, 11, 29], use meaning-based retrieval; recent developments suggest that hybrid techniques with fuzzy matching and time recency are possible [12, 28]. In terms of safety, voice automation often executes commands without constraints [21, 22]; Elghaish et al. [30] made an assistant to navigate construction documents but pointed out that the cloud computing of the service raises problems of latency and privacy. JSON command protocols keep the voice assistant systems easy to verify and integrate.

## 3 Research Gap and Problem Formulation

Although gaze tracking, speech recognition, and natural language processing have matured, a safe desktop file management system for individuals with upper-limb disabilities does not exist yet. We identify three deficiencies in the market for accessible technology. RG1 (multimodal synergy): each of the existing commercial solutions fails to integrate multiple modalities, while voice recession, for example, is ineffective on screens crowded with information, and gaze has false dwell time and Midas touch problems. RG2 (semantic and temporal retrieval): there are commercial tools for voice file management that do not realize the need to combine semantic mapping and time decay in order to perform successful prompting of the relevant commands to users. RG3 (verifiable safety): existing voice-activated personal assistants work heuristically or rely on the feedback from GUI prompts, which cannot be easily ignored by users. VDFM addresses these gaps through a unified multimodal interface, privacy-preserving edge ASR, a five-stage hybrid retrieval engine and a verified multi-tier safety mechanism.

## 4 Methodology and System Architecture

### 4.1 System Overview and End-to-End Multimodal Pipeline

VDFM is an offline-first, multimodal desktop interaction platform with three low-effort input channels: streaming speech, webcam dwell-gaze tracking and an external single switch. A local edge ASR engine transcribes spoken commands, a multi-stage natural language understanding (NLU) pipeline classifies the intent and extracts slots, and the result is serialized into a validated JSON command. Depending on the intent, a request goes to a context-aware hybrid retrieval engine (descriptive search) or to a filesystem operations engine (create, move, rename, copy, delete). A confidence-calibrated multi-tier safety layer holds destructive operations behind explicit

multimodal confirmation, and visual and auditory feedback closes the loop. Fig. 2 shows the architecture and dataflow.

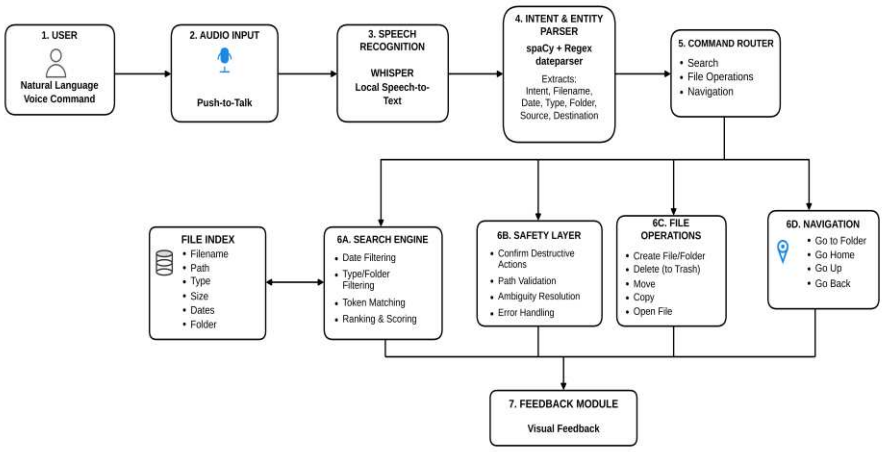


**Fig. 2** Overall architecture of the proposed voice-driven file management system.

The execution follows a deterministic flow. The user activates the system using a wake word or push-to-talk, gaze dwell, or physical switch. The voice activity detector segments the audio. The transcriptions are normalized and classified as canonical intents. The extraction module fills in the slots (source paths, destination folders, file types, and time filters). The Open Command Schema serializes the command. The safety policies check for danger before and after command execution.

### 4.2 Multimodal Input Acquisition, Fusion, and Triggering

- Voice Input and Acoustic Triggering: Audio is sampled at 16 kHz (16-bit mono PCM). Capture starts by push-to-talk or hands-free with openWakeWord [31] trained on the phrase 'Hey Assistant.' The Silero VAD [32] segments speech on 30 ms frames (probability threshold 0.60) and closes the buffer after an unvoiced pause longer than 750 ms.
- Webcam Gaze Tracking and Dwell Triggering: A module built on MediaPipe Face Mesh and Iris landmarks runs at 30 frames per second on a consumer webcam and maps fixations to screen coordinates after a 5-point calibration, with a mean angular error of 1.18° ± 0.22° (1.24 ± 0.23 cm, about 69 pixels, at 60 cm on a 15.6-inch Full HD display). To prevent accidental activation ('Midas touch'), file cards and confirmation dialogs require a continuous dwell (600–1200 ms, default 800 ms) with a circular progress timer for feedback.
- Single-Switch Interface: Standard accessibility switches (jelly buttons, micro-light switches, sip-and-puff devices) connect over USB HID or 3.5 mm relay. 50 ms hardware debouncing prevents spasm-induced double triggers, giving an activation latency of 186 ± 28 ms (Table 10). The switch can act as push-to-talk, target selection or single-click confirmation.
- Multimodal Fusion: The eye movement gives spatial anchoring (focusing on an unclear document), the uttered words explain the action ('Transfer that document to the Research Archive'), and the action or the verbal agreement provides the confirmation. This division of labor lowers cognitive load and lets users work through whichever channel tires them least.

### 4.3 Privacy-Preserving Edge Automatic Speech Recognition

File management involves sensitive documents and institutional data, so all speech recognition runs on the edge using Whisper [14], an encoder-decoder Transformer trained on 680,000 hours of weakly supervised multilingual audio that converts 16 kHz waveforms into 80-channel log-mel spectrograms (25 ms windows, 10 ms hop). We benchmark Whisper Tiny (39M parameters), Base (74M) and Small (244M) with INT8 quantization on an Intel Core i7-12700H CPU with 16 GB RAM (and an RTX 3060 Laptop GPU with 6 GB VRAM). Base takes 797 ± 64 ms and Small takes 1,624 ms (Table 3, Table 12); the deployed pipeline uses CPU INT8 so that it runs on laptops without a discrete GPU. Wireshark 4.2 monitoring detected no outbound TCP/UDP telemetry packets attributable to the evaluated speech, language and retrieval pipelines during recognition sessions under the tested configuration, which supports local operation but gives no guarantee about unmonitored background OS processes.

### 4.4 Natural Language Preprocessing and Canonical Intent Space

A closed set of 12 mutually exclusive intents, namely CREATE_FILE, CREATE_FOLDER, DELETE, DUPLICATE, RENAME, MOVE, COPY, OPEN, NAVIGATE, LIST, SEARCH, and UNKNOWN is created from normalized transcription. UNKNOWN is also assigned to out-of-domain commands to prevent any side effects. Table 1 provides the canonical verbs and mandatory slot entities for each intent.

**Table 1:** Semantic specification of the 12 file management intents, canonical verbs and required slot entities

| Intent Category | Canonical Operational Verbs | Required / Extracted Slot Entities |
|---|---|---|
| CREATE_FILE | create, make, new, generate, touch | source_name, file_type, target_directory |
| CREATE_FOLDER | create folder, make directory, mkdir | source_name, target_directory |
| DELETE | delete, remove, erase, trash, destroy | source_name, target_directory, file_type, target_scope, temporal_constraint |
| DUPLICATE | duplicate, clone, make copy, replicate | source_name, target_directory |
| RENAME | rename, change name to, call it | source_name, new_filename |
| MOVE | move, transfer, relocate, shift, cut | source_name, target_directory, file_type, target_scope |
| COPY | copy, replicate to, send copy to | source_name, target_directory, file_type, target_scope |
| OPEN | open, launch, view, read, display | source_name, file_type |
| NAVIGATE | go to, navigate to, open folder, cd | target_directory |
| LIST | list, show files, display contents, ls | target_directory, file_type, target_scope, temporal_constraint |
| SEARCH | search, find, locate, look for, where is | source_name, temporal_constraint, file_type, target_scope |
| UNKNOWN | out-of-domain conversational queries | none |

The terms “source_name,” “target_directory,” “new_filename” and “file_type” refer to file system targets, whereas the terms “temporal_constraint” and “target_scope” specify commands in batches such as “delete all temporary logs from last month.” As a statistical baseline, text commands are converted to numerical vectors with Term Frequency-Inverse Document Frequency (TF-IDF). For a token t in command document d within corpus D, the weight is:

$$\mathrm{TFIDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D) = \left(\frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}\right) \times \left[\log\left(\frac{1+|D|}{1+|\{d' \in D: t \in d'\}|}\right) + 1\right] \quad (1)$$

Using the method outlined above, we extract unigrams and bigrams as features. The method applies a coefficient that is less than 1 to term frequencies, normalizes values using L2 normalization, and retains verb-noun combinations like ‘create file’ and ‘search for’ while reducing the importance of frequently used stop words.

### 4.5 NLP-Based Intent Classification Models

- DELP (the baseline detector): This method utilizes a compilation engine to detect specific command prefixes and keywords, and it has a response time that is measured in sub-milliseconds. However, this method does not generalize and is not effective in handling paraphrasing and ASR-related errors.
- TF-IDF with Logistic Regression: This method operates by using the multinomial logistic regression approach (L2 regularization, C = 1.0) to associate the input TF-IDF vector x to a probability distribution of K = 12 intents through the softmax function.

$$P(Y = k \mid x) = \frac{exp(w_k \cdot x + b_k)}{\sum_{j=1}^{K} exp(w_j \cdot x + b_j)} \quad (2)$$

- Distilled RoBERTa Transformer: The unmodified RoBERTa model 26 has 6 encoder layers, 768 hidden dimensions, 12 attention heads, and 82 million parameters. [27] It is then fine-tuned on our command text corpus using AdamW. The parameters of the optimization algorithm are as follows: a learning rate of 2e-5 linear warmup over 10% of steps, a batch size of 16, cross-entropy, and 10 epochs.

### 4.6 Entity Extraction and Slot Filling

There are six types of slots identified, which are Source_File Name, Target_Directory Path, New_Filename, File_Type/Extension, Temporal/Date Constraint, and Action_Target Scope. We are using a hybrid of rule-based and dependency-parsing engines based on spaCy, which removes intent trigger tokens, identifies prepositional phrases introduced by the words 'to', 'into', 'in', 'from', and 'as', and takes tokens that follow the words 'named', 'called', and 'titled' as original file entities, as well as mapping extensions through an alias dictionary (e.g., spreadsheet to .xlsx). In addition to this, we resolve ambiguities by matching with local filesystem index, and if the required slot cannot be identified by the system, the dialogue manager requests the user to clarify the issue.

### 4.7 Open Command Schema and Interoperable API Architecture

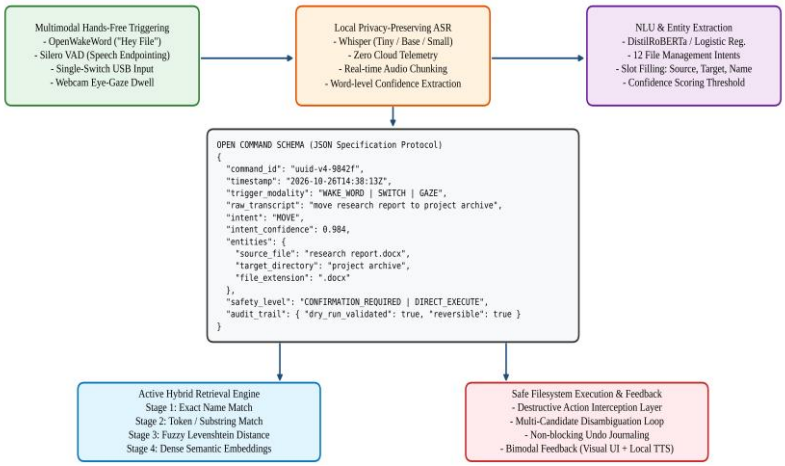


**Fig. 3** Documented Open Command Schema architecture and modular JSON protocol flow.

All authenticated multimodal commands are converted into a strictly verified JSON payload (Open Command Schema v2.1.0; Fig. 3) that includes the schema version, command ID, time, and trigger modality (wake word, voice activity detection, switch, or gaze dwell); transcription (raw text, automatic speech recognition model, acoustic confidence); intent (name, confidence, and classification); extracted entities; and safety evaluation (risk level, confirmation modality and requirement, timeout of ten seconds, reversibility).

The routing level delivers the confirmed instruction to a section of four subsystems: hybrid retrieval (SEARCH), filesystem (CREATE, DELETE, RENAME, MOVE, COPY, DUPLICATE, OPEN), navigation (NAVIGATE, LIST) or dialogue management (for questions of misunderstanding and security-related approvals). An easy-going local REST/IPC (Application Programming Interface) API (for example, /api/v2/command/execute, /api/v2/retrieval/query) provides an opportunity for the outside assistive software to send commands and receive formal telemetry data concerning the execution process.

### 4.8 File Indexing and Multi-Stage Hybrid Context-Aware Retrieval

An SQLite-reliant index, which remains up to date because of an automatic watcher (ReadDirectoryChangesW on Windows) retains the details of every file's path, filename, type, size, timestamps, and a 384-dimensional embedding e_d obtained from all-MiniLM-L6-v2 [11, 29]. The query q undergoes a five-step processing and ranking procedure:

- Stage 1—Exact Substring Match: case-insensitive presence of query tokens in the filename, S_exact(q, d) in {0, 1}.
- Stage 2—Token-Level Query Coverage: the fraction of query tokens found in the filename, which handles partial multi-word queries:

$$S_{token}(q,d) = \frac{|Tokens(q) \cap Tokens(d)|}{|Tokens(q)|} \quad (3)$$

- Stage 3—Fuzzy Normalized Levenshtein Similarity: character-level similarity that tolerates misspellings and ASR phonetic corruption:

$$S_{fuzzy}(q,d) = 1 - \frac{Levenshtein(q,d)}{max(|q|,|d|)} \quad (4)$$

- Stage 4—Dense Semantic Embedding: cosine similarity between query embedding e_q and file embedding e_d, which resolves conceptual descriptions that share no words with the filename:

$$S_{semantic}(q,d) = \frac{e_q \cdot e_d}{\| e_q \| \| e_d \|} \quad (5)$$

Stage 5—Context-Aware Hybrid Fusion and Temporal Filtering: time acts as a hard filter when the user states a date reference ('modified last week') and as a soft recency decay when a search is ambiguous. The hybrid score is S_hybrid(q, d) = (α S_exact + β S_token + γ S_fuzzy + δ S_semantic) × Γ(q, d) × T_recency(d):

$$S_{hybrid}(q,d) = (\alpha\, S_{exact}(q,d) + \beta\, S_{token}(q,d) + \gamma\, S_{fuzzy}(q,d) + \delta\, S_{semantic}(q,d)) \times \Gamma(q,d) \times T_{recency}(d) \quad (6)$$

The weights α = 0.30, β = 0.20, γ = 0.20 and δ = 0.30 were calibrated by grid search on 30 pilot queries kept separate from the 150 evaluation queries. The function Γ(q, d) takes a value of 1 for the 38 queries where the temporal expression is available and acquires a value of 1.0 in the other cases. The function T_recency(d) is defined as T_recency(d) = exp(- λ Δt_d), where Δt_d represents the number of days since a certain modification took place, λ is set to 0.005 which gives a half-life of around 138.6 days (i.e., it allows to cover both the semester cycle and quarterly cycle), which means that, for example, a file that has been modified a year ago receives a value of approximately 0.161. There were three annotators who assessed relevance (Fleiss' κ = 0.88). Out of a total of 150 queries there are 50 exact queries, 50 partial queries and 50 descriptive queries.

### 4.9 Filesystem Operations and Navigation Engine

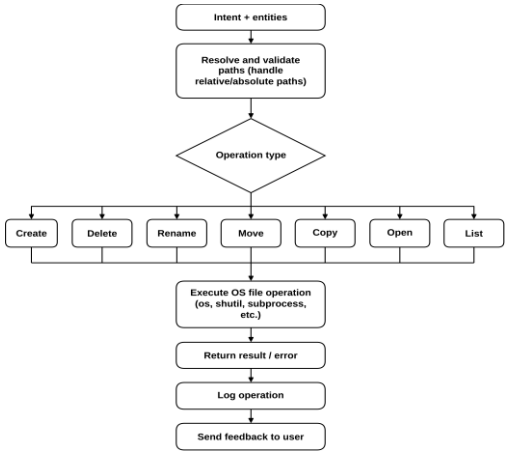


**Fig. 4** File operation execution, path validation, and multimodal feedback workflow.

According to Fig. 4, the operation process is illustrated, which sends validated commands through the APIs that belong to the OS. Throughout this process, all paths undergo normalizing and sanitizing, relative paths are calculated with the use of the specified working directory, path traversal ('..') as well as the system directories that

are protected are restricted, whilst any conflicts in terms of destination would result in either formatting the version by adding the suffix or solving the problem interactively.

### 4.10 Multi-Tier Safety and Confirmation Mechanism

A multipronged guardrail protects against incorrect speech recognition and unintentional triggers. In tier 1 (read-only: SEARCH, LIST, OPEN, NAVIGATE, UNKNOWN), a command is executed right away. In tier 2 (CREATE_FILE, CREATE_FOLDER, COPY, DUPLICATION), commands provide non-blocking HUD confirmations but ask for confirmation only in case of file path issues or ambiguous designation. The tier 3 (DELETE, multi-directory MOVE, RENAME, and other destructive commands) is always stopped by a deterministic policy and the prompt itself states the word "verb," specified destination file name, and chosen location.

Confirmation methods correspond with the user's capabilities and can be done either by words ('Confirm'/'Cancel'), by resting eyes for 800 ms on the screen button, or by tapping the button. If a command is not confirmed within 10 seconds, it is stopped, and files are moved to the OS Recycle Bin for recovery purposes.

### 4.11 Multimodal User Feedback

When users encounter visual fatigue, synchronized bimodal feedback comes to their aid: a text-to-speech engine (pyttsx3 / Windows SAPI5) provides brief status updates (for instance, 'Found 3 files matching your request'), while a translucent high-resolution visor displays the text, intention categorization, information about matching files, and affirmative response buttons with radial timer indicators.

### 4.12 Experimental Methodology and Evaluation Design

#### *4.12.1 Dataset Preparation and Partitioning Protocol*

To avoid burdening participants with severe motor impairments, the command corpus was generated synthetically in three phases: canonical syntactic seeding (templates per intent combining imperative verbs, polite prefaces and indirect questions), lexical entity injection (filenames, folder paths, extensions and temporal modifiers sampled from curated lexicons) and paraphrasing with controlled noise (contractions, word-order inversions, ellipses and fillers). The 1,440 commands (120 per intent) were split into 1,008 training, 216 validation and 216 independent test commands (84/18/18 per intent) and additionally evaluated with stratified 5-fold cross-validation over the full corpus. Because template corpora saturate (99.5–100%), generalization is tested on real speech: a 72-sample live manifest (6 speakers, 12 commands each, quiet office, SNR ~32 dB, Rode NT-USB microphone) and an independent 240-recording held-out benchmark (8 speakers; General American, British RP, Indian and Australian English, 60 recordings each; clean >30 dB, office ~15 dB and ambient ~5 dB SNR). A template-disjoint test on 120 unseen paraphrases gave 97.50% intent accuracy.

#### *4.12.2 Evaluation Metrics and Statistical Formulations*

Intent classification uses overall accuracy and macro-averaged precision, recall and F1-score:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i) \tag{7}$$

$$Precision_{macro} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FP_c} \tag{8}$$

$$Recall_{macro} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FN_c} \tag{9}$$

$$F1_{macro} = 2 \times \frac{Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{10}$$

ASR accuracy uses Word Error Rate (WER) and Character Error Rate (CER) from Levenshtein alignment, reported as raw WER and as normalized WER using the Whisper English text normalizer:

$$WER = \frac{S_{words} + D_{words} + I_{words}}{N_{words}} \tag{11}$$

$$CER = \frac{S_{chars} + D_{chars} + I_{chars}}{N_{chars}} \tag{12}$$

Command-Weighted WER (CW-WER): misrecognizing an operational verb ('delete' vs 'create') or a target filename is serious, whereas dropping a filler ('please') has no operational effect. Building on WIL [33], SemER [34], Task-Oriented WER [35] and task-based ASR evaluation [36], CW-WER weights substitution (S), deletion (D) and insertion (I) errors by an operational category weight omega(w):

$$\mathrm{CW-WER} = \frac{\sum_{w\in S}\omega(w) + \sum_{w\in D}\omega(w) + \sum_{w\in I}\omega_{\mathrm{ins}}(w)}{\sum_{w\in \mathrm{Ref}}\omega(w)} \tag{13}$$

The weights are w_op = 0.45 (action verbs), w_file = 0.35 (file entities and paths), w_mod = 0.15 (syntactic modifiers) and w_fill = 0.05 (fillers); insertions take the weight of their category, or w_fill if extraneous. The agreement among three annotators on slots and intents was kappa = 0.91 (Fleiss). Confidence intervals use the Student's t-distribution (t_0.025,7 = 2.365 for the N = 8 accessibility cohort):

$$CI_{95\%} = \bar{x} \pm t_{\alpha/2,n-1} \times \frac{s}{\sqrt{n}} \tag{14}$$

Retrieval is evaluated with Mean Reciprocal Rank (MRR) [37] and Recall@k (k in {1, 3, 5}), where rank_i is the rank of the first relevant document for query i:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{15}$$

#### *4.12.3 Human-Centered Pilot Usability Protocol*

Ethics and Participants: The Institutional Ethics Committee (IEC-VIT) approved the protocol (VIT/IEC/2026/04-A, 14 April 2026), all procedures followed the 1964 Declaration of Helsinki, and written informed consent was obtained (witnessed verbal or digital consent where impairment prevented signing). Twelve participants took part: an accessibility cohort of eight people (P1–P8; 5 male, 3 female; aged 21–64) with clinically diagnosed cerebral palsy, C5–C6 spinal cord injury, muscular dystrophy or severe tremor/neurodegenerative impairment (two per group), and four able-bodied controls (P9–P12) who set normative baselines. All inferential statistics use only the eight-person accessibility cohort. Formal dysarthria severity profiling was outside the scope of this pilot (Section 6).

Task Battery: Each participant performed six tasks: T1 create the directory 'Conference Papers' in Documents; T2 search for and open 'Research_Proposal_v2.docx'; T3 locate a spreadsheet of last quarter's travel budget without knowing its filename (descriptive search); T4 move 'Seminar_Slides.pptx' from Downloads to 'Presentations'; T5 rename 'Draft_Manuscript_2026.docx' to 'Final_Camera_Ready.docx'; and T6 locate and delete 'Old_Temporary_Log.txt' and confirm the safety prompt.

Design and Questionnaires: A within-subjects design compared Keyboard/Mouse (KM), Windows Voice Access (WVA) and VDFM, with condition order counterbalanced by a balanced Latin square after a 10-minute familiarization session. In the VDFM condition participants used the configuration that suited them: gaze dwell (800 ms) + voice or hands-free voice (P3–P6) or switch + voice (P1, P2, P7, P8). In the KM baseline, participants with severe paralysis used their personal assistive devices (head pointer, trackball with caregiver help). Tasks timed out at 180 s; reported times are for completed trials, and Section 5.6 gives a sensitivity analysis with timeouts capped at 180 s. After each condition, participants completed the NASA Task Load Index (NASA-TLX) [38] (six subscales, 0–100, unweighted Raw TLX) and the 10-item System Usability Scale (SUS) [39].

Statistical Analysis: Two-tailed paired t-tests (t(7)) compared completion times, composite workload (RTLX), SUS and the six NASA-TLX subscales after Shapiro-Wilk normality checks; paired Wilcoxon signed-rank tests served as sensitivity analysis (minimum attainable $p = 0.0078$ for $N = 8$). Effect sizes are Cohen's $d_z$ with 95% CIs, and Bonferroni correction across the five primary comparisons (times VDFM vs KM and WVA, RTLX vs KM, SUS vs KM and WVA) gives $\alpha_{adj} = 0.05 / 5 = 0.010$.

#### *4.12.4 Dedicated Multimodal Interaction Benchmarking Protocol*

All eight motor-impaired participants compared four input configurations: Voice-Only (push-to-talk, requiring finger pressing), Voice + Wake Word (hands-free), Gaze-Dwell + Voice (hands-free) and Switch + Voice (gross movement only). Each completed 4 disambiguation trials per modality on crowded or confusable targets (e.g., 'report_v1.docx' vs 'report_v2.docx'; 32 trials per modality). False triggers were logged over 16 exposure hours per modality (64 modality-hours), and a forced-choice survey recorded preference. Output feedback (HUD only, TTS only, synchronized audio-visual) was benchmarked for latency, perception accuracy and preference.

## 5 Results and Discussion

### 5.1 Intent Classification Performance

Table 2 reports the classifiers on the independent 216-command test split and under 5-fold cross-validation. DistilRoBERTa reached 100.00% (216/216) on the test split and 99.86% ± 0.19% in cross-validation (1,438 of 1,440 correct; fold accuracies 100.00, 99.65, 100.00, 100.00 and 99.65%). It outperformed TF-IDF + Logistic Regression (99.54%), which labeled 'make another version of seminar notes' as CREATE_FILE because of the weight on 'make', whereas DistilRoBERTa returned DUPLICATE by capturing the context of 'another version'. The regex baseline (93.52%, 202/216) failed on conversational prefaces, elliptical syntax and inverted word order, with its largest confusions between MOVE and COPY (5), CREATE_FILE and CREATE_FOLDER (4) and SEARCH and OPEN (3).

**Table 2:** Intent classification on the independent test split and under 5-fold cross-validation

| Classification Model Architecture | Evaluation Setup / Split | Overall Accuracy (%) | Macro Precision (%) | Macro Recall (%) | Macro F1-Score (%) |
|---|---|---|---|---|---|
| Deterministic Regex Parser (Baseline) | Independent Test Split (N=216) | 93.52% (202 / 216) | 94.98% | 93.52% | 93.54% |
| TF-IDF + Multinomial Logistic Regression | Independent Test Split (N=216) | 99.54% (215 / 216) | 99.56% | 99.54% | 99.54% |
| Fine-Tuned DistilRoBERTa Transformer | Independent Test Split (N=216) | 100.00% (216 / 216) | 100.00% | 100.00% | 100.00% |
| TF-IDF + Multinomial Logistic Regression | Stratified 5-Fold Cross-Validation | 99.44% ± 0.38% | 99.48% ± 0.35% | 99.44% ± 0.38% | 99.44% ± 0.37% |
| Fine-Tuned DistilRoBERTa Transformer | Stratified 5-Fold Cross-Validation | 99.86% ± 0.19% | 99.87% ± 0.18% | 99.86% ± 0.19% | 99.87% ± 0.18% |

### 5.2 Comparative ASR Benchmarking and Command-Weighted Error Analysis

We benchmarked Whisper Tiny, Base and Small on the 72-command speech manifest (Table 3). Tiny is the fastest (0.382 s), but its normalized CW-WER of 15.60% lowers downstream intent accuracy to 86.11% with TF-IDF + LR. Small is most accurate (normalized WER 9.80%, CW-WER 6.20%, 95.83%), but its 1.624 s latency is noticeable on low-power devices. Base gives the best balance: 0.797 s latency, under 600 MB RAM and 90.28% intent accuracy with TF-IDF + LR (93.06% with the deployed DistilRoBERTa; 97.22% on Small; clean text 100.00%). Base's raw WER of 44.86% is inflated by capitalization, punctuation and compound splits; normalization lowers it to 14.20% and normalized CW-WER is 9.80%, because errors fall mostly on fillers ('please') or compounds ('textfile'), not on verbs or filenames. Normalized CW-WER correlates much more strongly with downstream intent accuracy ($r_{pb} = -0.88$, 95% CI [−0.92, −0.81]) than normalized WER ($r = -0.58$; Williams' $t(69) = 9.07$, Meng's $Z = 6.68$, $p < 0.0001$), and this was stable when the keyword-to-filler weight ratio varied from 2.0 to 5.0 ($r_{pb}$ between −0.85 and −0.91).

**Table 3:** ASR performance across Whisper variants on 72 core spoken commands

| ASR Model Architecture | Model Parameters | Raw Corpus WER (%) | Normalized WER (%) | Corpus CER (%) | Normalized CW-WER (%) | Downstream Intent: TF-IDF / DistilRoBERTa (%) | Mean Inference Latency (s) |
|---|---|---|---|---|---|---|---|
| Whisper Tiny | 39 Million | 54.21% | 22.40% | 13.82% | 15.60% | 86.11% (62/72) / 90.28% (65/72) | 0.382 s |
| Whisper Base (Proposed) | 74 Million | 44.86% | 14.20% | 10.34% | 9.80% | 90.28% (65/72) / 93.06% (67/72) | 0.797 s |
| Whisper Small | 244 Million | 28.14% | 9.80% | 6.18% | 6.20% | 95.83% (69/72) / 97.22% (70/72) | 1.624 s |

### 5.3 Robustness on Held-Out Data: Accents and Acoustic Noise

On the 240-recording held-out benchmark (Table 4), Whisper Base kept 84.58% intent accuracy (95% CI [79.5%, 88.6%]; 203/240) and Whisper Small kept 92.08% ([88.0%, 94.9%]; 221/240). Accuracy decreased in the presence of noise (Base: 91.25% for clean signal, 86.25% at 15 dB, 76.25% at 5 dB; Small: 96.25%, 93.75%, 86.25%) and showed variations due to the accent (Base: 90.00% General American, 86.67% British RP, 81.67% Indian, 80.00% Australian; Small: 95.00%, 93.33%, 91.67%, 88.33%), although the Base's CW-WER (12.30% in total) is far less than its unprocessed WER (52.90%).

**Table 4:** Whisper Base and Small across accents and noise tiers on the 240-recording held-out benchmark

| Evaluation Stratum / Acoustic Condition | Sample Size (N) | Whisper Base Raw WER (%) | Whisper Base Normalized WER (%) | Whisper Base Normalized CW-WER (%) | Downstream Intent: Base (%) | Downstream Intent: Small (%) |
|---|---|---|---|---|---|---|
| Clean Studio Audio (SNR > 30 dB) | 80 | 42.10% | 13.40% | 9.10% | 91.25% [83.0%, 95.7%] | 96.25% [89.5%, 98.7%] |
| Office Noise (SNR ~ 15 dB: typing + HVAC) | 80 | 52.40% | 16.80% | 11.60% | 86.25% [77.0%, 92.1%] | 93.75% [86.2%, 97.3%] |
| Ambient Noise (SNR ~ 5 dB: cafeteria chatter) | 80 | 64.20% | 22.50% | 16.20% | 76.25% [65.9%, 84.2%] | 86.25% [77.0%, 92.1%] |
| Regional Accent: General American | 60 | 48.20% | 15.10% | 10.40% | 90.00% [79.9%, 95.3%] | 95.00% [86.3%, 98.3%] |
| Regional Accent: British Received Pronunciation | 60 | 50.80% | 16.50% | 11.60% | 86.67% [75.8%, 93.1%] | 93.33% [84.1%, 97.4%] |
| Regional Accent: Indian English | 60 | 55.40% | 18.80% | 13.20% | 81.67% [70.1%, 89.4%] | 91.67% [81.9%, 96.4%] |
| Regional Accent: Australian English | 60 | 57.20% | 19.90% | 14.00% | 80.00% [68.2%, 88.2%] | 88.33% [77.8%, 94.2%] |
| **Consolidated Held-Out Benchmark** | **240** | **52.90%** | **17.57%** | **12.30%** | **84.58% [79.5%, 88.6%]** | **92.08% [88.0%, 94.9%]** |

### 5.4 Entity Extraction and Safety Mechanism Validation

On the 216 test utterances, entity extraction reached a micro-F1 of 97.02% (precision 97.75%, recall 96.31%) over 406 slot tokens and a complete-frame accuracy of 93.06% (201/216) (Table 5). File extensions (F1 = 98.61%) and source names (97.28%) were best; temporal constraints were weakest (93.55%), with lower recall on colloquial relative expressions ('from the start of the semester'). The safety mechanism was tested on 100 scenarios, presented by voice under 15 dB of office noise and by synthetic injection of acoustically distorted destructive commands (Table 6). All 30 Tier 3 scenarios were intercepted (100%, Wilson 95% CI [88.65%, 100%]). In Tier 2, 1 of 30 scenarios triggered an unnecessary confirmation on an ambiguous path (3.33%; 1.43% across the 70 benign scenarios), Tier 1 had none, and no unintended file mutations occurred in any scenario (0/100, Wilson upper bound 3.7%).

**Table 5** Entity extraction performance across semantic slot types

| Semantic Slot Type | Ground Truth Tokens | Slot Precision (%) | Slot Recall (%) | Slot F1-Score (%) |
|---|---|---|---|---|
| Source_File Name | 148 | 97.94% | 96.62% | 97.28% |
| Target_Directory Path | 82 | 96.34% | 96.34% | 96.34% |
| New_Filename (Rename) | 18 | 100.00% | 94.44% | 97.14% |
| File_Type / Extension | 72 | 98.61% | 98.61% | 98.61% |
| Temporal / Date Constraint | 32 | 96.67% | 90.62% | 93.55% |
| Action_Target Scope | 54 | 98.11% | 96.30% | 97.20% |
| Overall Micro-Average (All Slots) | 406 | 97.75% | 96.31% | 97.02% |
| Complete Frame Accuracy (All Slots Correct) | 216 commands | — | — | 93.06% (201 / 216) |

**Table 6:** Multi-tier safety and confirmation mechanism across 100 scenarios (Wilson 95% CI)

| Operational Scenario Risk Tier | Total Scenarios Tested | Safety Interception Rate (95% CI) | Safe Resolution Rate (%) | Unintended Mutations (95% CI) |
|---|---|---|---|---|
| Tier 1: Read-Only / Inspection (5 Intents) | 40 | 0.0% (0 / 40) [0.0%, 8.8%] | 100.0% (40 / 40) | 0.0% (0 / 40) [0.0%, 8.8%] |
| Tier 2: Benign Reversible Modifications (4 Intents) | 30 | 3.33% (1 / 30) [0.6%, 16.7%] | 100.0% (30 / 30) | 0.0% (0 / 30) [0.0%, 11.3%] |
| Tier 3: Destructive / State Mutation (3 Intents) | 30 | 100.0% (30 / 30) [88.65%, 100%] | 100.0% (30 / 30) | 0.0% (0 / 30) [0.0%, 11.3%] |
| **Overall Guardrail Reliability Across All Tiers** | **100** | **31.00% (31 / 100) [22.8%, 40.7%]** | **100.0% (100 / 100)** | **0.00% (0 / 100) [0.0%, 3.7%]** |

### 5.5 Context-Aware Retrieval Evaluation and Component Ablation

The testbed held 2,500 files and 150 queries (50 exact, 50 partial and 50 descriptive). Table 7 and Fig. 5 isolate each component. On descriptive queries, exact matching fails almost completely (MRR = 0.018) and BM25 reaches only 0.245 (Recall@5 = 36.0%); dense embeddings alone raise MRR to 0.742 (Recall@5 = 88.0%) and all-query MRR to 0.859, confirming that they resolve descriptions with no lexical overlap. Temporal filtering alone raised MRR on the 38 temporal queries from 0.452 to 0.684, but only +0.029 overall (0.646 to 0.675) because recency decay slightly penalizes older non-temporal files when no semantic signal anchors relevance. The full hybrid engine reaches an MRR of 0.893 and Recall@5 of 96.67% overall (0.818 and 94.00% on descriptive queries), significantly outperforming BM25 + Dense with Reciprocal Rank Fusion ($Z = 4.18$, $p < 0.001$) and the Windows Search Indexer ($Z = 8.85$, $p < 0.001$).

**Table 7:** Multi-stage ablation of hybrid context-aware retrieval (selected stages and baselines)

| Retrieval Pipeline Stage / Method | Query Category Evaluated | Recall@1 (%) | Recall@3 (%) | Recall@5 (%) | Mean Reciprocal Rank (MRR) |
|---|---|---|---|---|---|
| Commercial Baseline: Windows Search Indexer | All Combined (N=150) | 36.67% | 42.00% | 44.67% | 0.413 |
| Stage 1: Exact Substring Only | All Combined (N=150) | 33.33% | 36.67% | 38.00% | 0.361 |
| Stage 2: + Token Query Coverage | All Combined (N=150) | 48.67% | 58.00% | 60.67% | 0.542 |
| Stage 3: + Fuzzy Levenshtein Distance | All Combined (N=150) | 58.00% | 68.00% | 72.67% | 0.646 |
| Standard Lexical BM25 Baseline | Descriptive Queries (N=50) | 18.00% | 28.00% | 36.00% | 0.245 |
| Standard Lexical BM25 Baseline | All Combined (N=150) | 54.67% | 64.00% | 68.67% | 0.612 |
| Stage 4a: + Dense Embeddings Alone (No Temporal) | Descriptive Queries (N=50) | 64.00% | 82.00% | 88.00% | 0.742 |
| Stage 4a: + Dense Embeddings Alone (No Temporal) | All Combined (N=150) | 79.33% | 90.67% | 94.00% | 0.859 |

| Retrieval Pipeline Stage / Method | Query Category Evaluated | Recall@1 (%) | Recall@3 (%) | Recall@5 (%) | Mean Reciprocal Rank (MRR) |
|---|---|---|---|---|---|
| BM25 + Dense with Reciprocal Rank Fusion (RRF) | All Combined (N=150) | 80.00% | 91.33% | 94.67% | 0.868 |
| Stage 4b: Temporal Filtering Alone (38 Temporal Queries) | Temporal Subset (N=38) | 60.53% | 71.05% | 76.32% | 0.684 |
| Stage 4b: Temporal Filtering Alone (No Dense) | All Combined (N=150) | 61.33% | 71.33% | 76.00% | 0.675 |
| Stage 5: Full Proposed Hybrid (Dense + Temporal) | Descriptive Queries (N=50) | 72.00% | 88.00% | 94.00% | 0.818 |
| Stage 5: Full Proposed Hybrid (Dense + Temporal) | All Combined (N=150) | 82.67% | 93.33% | 96.67% | 0.893 |

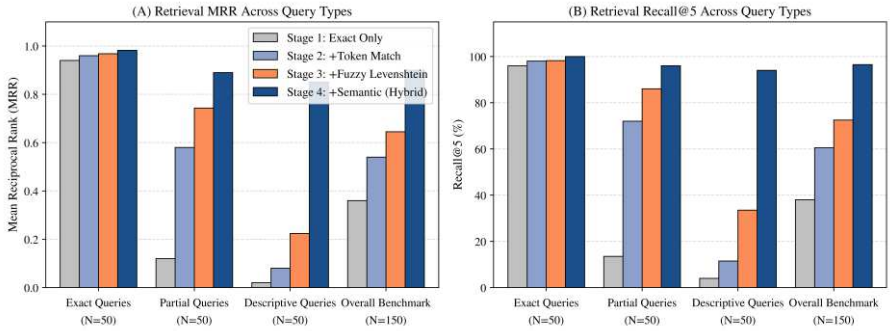


**Fig. 5** Retrieval performance ablation across exact, partial, and descriptive query categories.

### 5.6 Human-Subject User Study: Task Performance, Workload, and Usability

Table 8, Table 9, Fig. 6 and Fig. 7 summarize the pilot study for the accessibility cohort (N = 8). VDFM cut mean task time from 78.41 ± 13.06 s (KM) to 24.60 ± 4.42 s, a 68.63% reduction ($t(7) = 14.98$, $p < 0.001$, $d_z = 5.30$, 95% CI [−62.31, −45.31] s), and was 46.76% faster than WVA (46.21 ± 8.87 s; $t(7) = 9.32$, $p < 0.001$). Timeouts occurred in 11/48 KM, 13/48 WVA and 2/48 VDFM trials; with timeouts imputed at 180 s, mean times were 31.08 s (VDFM), 101.70 s (KM) and 82.45 s (WVA), so excluding timeouts actually favors the baselines. On descriptive search (T3), WVA completed only 50.0% because lexical search failed on conceptual phrasing, whereas VDFM completed 87.5% (95.83% across all 48 trials). All six NASA-TLX subscales fell significantly (all $p < 0.001$; Wilcoxon $W = 0$, $p = 0.0078$): physical demand dropped 75.08% (74.25 to 18.50), frustration 65.39% and effort 60.88%, and Raw TLX fell 61.68% (60.42 to 23.15; $t(7) = 21.30$, $d_z = 7.53$). SUS reached 86.88 ± 5.13 (Grade A) versus 48.44 for KM (Grade F) and 61.25 for WVA (Grade D), above the 68.0 benchmark and the 80.3 cut-off for the 90th percentile. All five primary comparisons remained significant after Bonferroni correction ($p < 0.001 < 0.010$).

**Table 8** Pilot study cohort results: completion rate, time and error recovery by interaction modality

| Participant / Cohort Stratum | Interaction Modality / Baseline Comparison | Task Completion Rate (%) | Mean Completion Time (s) | Time Reduction vs KM (%) | Mean Error Recovery Cycles / Task |
|---|---|---|---|---|---|
| Accessibility Cohort Mean (N=8) | Keyboard & Mouse (KM) | 77.08% (37 / 48) | 78.41 ± 13.06 s | Baseline (0.0%) | 3.88 ± 0.99 cycles |
| Accessibility Cohort Mean (N=8) | Windows Voice Access (WVA) | 72.92% (35 / 48) | 46.21 ± 8.87 s | 41.07% faster | 2.75 ± 0.71 cycles |
| Accessibility Cohort Mean (N=8) | Proposed VDFM Assistant | 95.83% (46 / 48) | 24.60 ± 4.42 s | 68.63% faster | 0.62 ± 0.52 cycles |
| Control Cohort Mean (N=4) | Keyboard & Mouse (KM) | 100.0% (24 / 24) | 22.80 ± 4.60 s | Baseline (0.0%) | 0.50 ± 0.58 cycles |
| Control Cohort Mean (N=4) | Windows Voice Access (WVA) | 87.50% (21 / 24) | 38.50 ± 6.70 s | +68.86% longer | 2.25 ± 0.50 cycles |
| Control Cohort Mean (N=4) | Proposed VDFM Assistant | 95.83% (23 / 24) | 19.40 ± 3.80 s | 14.91% faster | 0.25 ± 0.50 cycles |

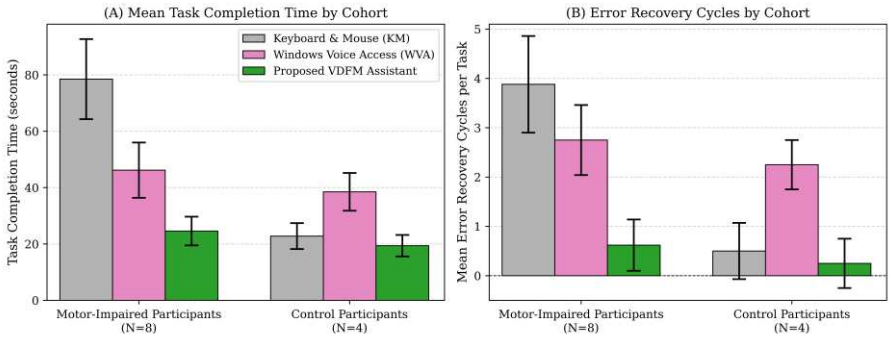


**Fig. 6** User study task completion time and error recovery cycles across cohorts.

**Table 9:** SUS and six-dimensional NASA-TLX workload from participant-level paired observations

| Evaluation Metric / Subscale | Keyboard & Mouse (KM) | Windows Voice Access (WVA) | Proposed VDFM Assistant | Paired t-test (VDFM vs KM) | Cohen's d_z Effect Size | 95% CI of Difference |
|---|---|---|---|---|---|---|
| NASA-TLX: Mental Demand (0–100) | 48.50 ± 6.26 | 54.25 ± 6.96 | 28.38 ± 3.62 | t(7) = 20.33, p < 0.001 | d = 7.19 | [−22.45, −17.80] |
| NASA-TLX: Physical Demand (0–100) | 74.25 ± 9.94 | 32.50 ± 6.07 | 18.50 ± 3.82 | t(7) = 24.60, p < 0.001 | d = 8.70 | [−61.08, −50.42] |
| NASA-TLX: Temporal Demand (0–100) | 58.00 ± 8.80 | 49.62 ± 7.42 | 24.25 ± 4.30 | t(7) = 20.25, p < 0.001 | d = 7.16 | [−37.70, −29.80] |
| NASA-TLX: Performance (Dissatisfaction: 0–100) | 51.25 ± 8.01 | 43.75 ± 7.27 | 19.50 ± 3.42 | t(7) = 18.70, p < 0.001 | d = 6.61 | [−35.78, −27.72] |
| NASA-TLX: Effort (0–100) | 68.38 ± 8.91 | 51.00 ± 8.16 | 26.75 ± 4.77 | t(7) = 27.33, p < 0.001 | d = 9.66 | [−45.21, −38.04] |
| NASA-TLX: Frustration (0–100) | 62.12 ± 11.23 | 48.25 ± 8.84 | 21.50 ± 4.07 | t(7) = 16.21, p < 0.001 | d = 5.73 | [−46.56, −34.69] |

| Evaluation Metric / Subscale | Keyboard & Mouse (KM) | Windows Voice Access (WVA) | Proposed VDFM Assistant | Paired t-test (VDFM vs KM) | Cohen's d_z Effect Size | 95% CI of Difference |
|---|---|---|---|---|---|---|
| NASA-TLX: Overall Raw TLX (RTLX: 0–100) | 60.42 ± 8.88 | 46.56 ± 7.49 | 23.15 ± 3.95 | t(7) = 21.30, p < 0.001 | d = 7.53 | [−41.41, −33.13] |
| System Usability Scale (SUS: 0–100) | 48.44 ± 4.54 (Grade F) | 61.25 ± 7.93 (Grade D) | 86.88 ± 5.13 (Grade A) | t(7) = 17.46, p < 0.001 | d_z = 6.17 | [+33.23, +43.65] |

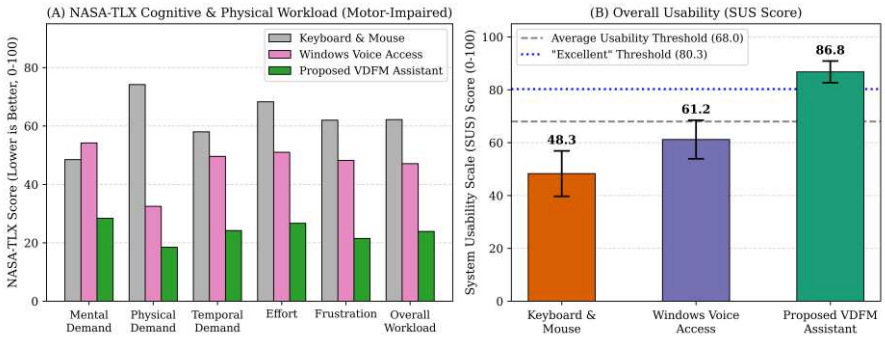


**Fig. 7** NASA-TLX subjective workload subscales and System Usability Scale (SUS) ratings.

### 5.7 Dedicated Multimodal Interaction Benchmark: Input Modalities and Feedback

Table 10 compares the input modalities. Gaze-Dwell + Voice gave the highest disambiguation accuracy (96.88%, 31/32) with the fastest hands-free execution (25.66 ± 5.50 s) and was the most preferred (37.5%, mainly users with quadriplegia or muscular dystrophy): fixating on a file card while saying 'Delete this file' avoids speaking disambiguating paths. The debounced switch gave the lowest activation latency (186 ms) with no false triggers, suiting participants with spastic cerebral palsy or severe tremor. Voice + Wake Word and Gaze-Dwell + Voice are completely hands-free. For output (Table 11), synchronized audio-visual feedback achieved 100% status perception (32/32) and was preferred by 87.5% of participants: speech spares users from looking away, and the HUD gives lasting reassurance during confirmations.

**Table 10:** Input modality comparison in the motor-impaired cohort (16 exposure hours per modality)

| Interaction Input Modality | Input Trigger Mechanism | Physical Action Required | Activation Latency (ms) | False Trigger Rate (per hour; 16h/modality) | Disambiguation Success (N=32 Trials) | Mean Task Completion Time (s) | Participant Preference (% of Cohort) |
|---|---|---|---|---|---|---|---|
| Voice-Only (Push-to-Talk) | Keyboard spacebar / mouse click | Physical key press required | 342 ± 45 ms | 0.00 / hr (0 / 16 hrs) | 78.13% (25 / 32) | 31.40 ± 6.2 s | 12.5% (1 / 8) |
| Voice + Wake Word | Acoustic 'Hey Assistant' (openWakeWord) | Completely hands-free | 418 ± 52 ms | 0.38 / hr (6 / 16 hrs) | 81.25% (26 / 32) | 28.50 ± 5.8 s | 25.0% (2 / 8) |
| Gaze-Dwell + Voice (Proposed) | Webcam iris dwell (800 ms) + Voice command | Completely hands-free | 824 ± 68 ms | 0.25 / hr (4 / 16 hrs) | 96.88% (31 / 32) | 25.66 ± 5.50 s | 37.5% (3 / 8) |
| Switch + Voice (Assistive Switch) | Debounced mechanical switch click | Gross physical movement required | 186 ± 28 ms | 0.00 / hr (0 / 16 hrs) | 84.38% (27 / 32) | 25.20 ± 4.9 s | 25.0% (2 / 8) |

**Table 11:** Output feedback modalities: latency, status perception accuracy and preference

| Feedback Modality Configuration | Feedback Delivery Channel | Mean Delivery Latency (ms) | Status Perception Accuracy (N=32 Trials) | Participant Usability Preference (% of Cohort) |
|---|---|---|---|---|
| Visual HUD Only | Top-anchored translucent desktop notification widget | 85 ± 12 ms | 87.50% (28 / 32) | 0.0% (0 / 8) |
| Auditory TTS Only | Offline synthetic speech vocalization (pyttsx3) | 128 ± 20 ms | 90.62% (29 / 32) | 12.5% (1 / 8) |
| Synchronized Audio-Visual (Proposed) | Synchronized desktop HUD notification + earcon chime with concurrent TTS vocalization | 112 ± 18 ms | 100.0% (32 / 32) | 87.5% (7 / 8) |

### 5.8 Edge Resource Footprint, Offline Verification, and Privacy

Table 12 profiles resource use. Isolated component allocations sum to 1,318 MB RAM, but shared runtimes and memory-mapped weights bring the integrated CPU-only working set to 820 MB at peak (18.5% mean CPU). End-to-end latency from speech endpoint to action is 937 ± 78 ms: VAD 18 ms, gaze tracking 28 ms, Whisper Base 797 ms, intent parsing 34 ms, slot extraction 12 ms, retrieval 42 ms and execution with safety checks 6 ms (909 ms without gaze). Because speech, language and retrieval run locally, no outbound telemetry attributable to them was observed; however, local processing does not by itself ensure compliance with regulatory frameworks such as HIPAA, which still depends on host security policies, disk encryption and access privileges.

**Table 12** Edge resource utilization and latency per pipeline subsystem (937 ms: measured gaze-assisted CPU pipeline; 909 ms: component sum without gaze)

| Subsystem / Pipeline Component | Underlying Technology / Model | Mean CPU Utilization (%) | Peak RAM Footprint (MB) | GPU VRAM Allocation (MB) | Execution Latency (ms) |
|---|---|---|---|---|---|
| Audio Capture & Voice Activity Detection | PyAudio + Silero VAD (ONNX) | 2.4% ± 0.6% | 84 MB | — (CPU execution) | 18 ± 4 ms / 30 ms frame |
| Webcam Gaze Tracking & Dwell Engine | MediaPipe Face Mesh + Iris (OpenCV) | 8.6% ± 1.4% | 142 MB | — (CPU/DirectML) | 28 ± 5 ms / frame (30 fps) |
| Speech Recognition (ASR) | Whisper Base (INT8 CPU / FP16 GPU) | 14.2% ± 2.1% | 540 MB | 890 MB (GPU cached) | 797 ± 64 ms (INT8 CPU) |
| Intent Classification & NLU | DistilRoBERTa (ONNX Runtime) | 3.8% ± 0.8% | 210 MB | 280 MB (GPU) | 34 ± 6 ms / command |
| Entity Extraction & Slot Parsing | spaCy Dependency Matcher | 1.5% ± 0.4% | 125 MB | — (CPU execution) | 12 ± 3 ms / command |
| Hybrid Semantic File Retrieval | SQLite + all-MiniLM-L6-v2 | 4.2% ± 0.9% | 185 MB | 160 MB (GPU) | 42 ± 8 ms / 2500 files |
| Safety Guardrail & System Execution | Native OS APIs + JSON Router | 0.8% ± 0.2% | 32 MB | — (CPU execution) | 6 ± 2 ms / command |
| **Consolidated System Active Execution** | **Gaze-Assisted Multimodal CPU Pipeline** | **18.5% ± 3.2%** | **820 MB** | **N/A (CPU-only)** | **937 ± 78 ms (Gaze-Assisted CPU)** |

## 6 Conclusion, Limitations, and Future Work

We presented VDFM, an offline-first multimodal file management system for users with upper-limb motor impairments that combines Whisper Base, MediaPipe dwell-gaze tracking and single-switch input. A fine-tuned

DistilRoBERTa classifies 12 intents with 100.00% test and 99.86% ± 0.19% cross-validated accuracy; spoken commands reach 93.06% intent accuracy on Whisper Base (84.58% on the held-out acoustic suite); slot extraction reaches a 97.02% micro-F1; five-stage hybrid retrieval reaches an MRR of 0.893; and the safety mechanism intercepted all destructive operations (30/30). During the pilot, VDFM completed a task in 68.63% less time and reduced the amount of work by 61.68% with a SUS score of 86.88, indicating an easy-to-use, private and very usable desktop interface.

Limitations: the pilot size was very limited (eight subjects) and its significant results reflect the degree of difficulty in using a mouse; the 1440-command database is synthetic and saturates, therefore, real free speech remains to be studied; dysarthria was not examined; the system only supports English and 12 commands and was tested on Windows 11 mainly; gaze tracking fails in poor lighting conditions or with thick glasses; transcription may take longer than 2 seconds on low-spec devices; and the research is a test in a controlled lab environment.

Future work will pursue (1) LoRA-based adaptation of Whisper on dysarthric speech corpora such as TORGO [16] and Project Euphonia [15]; (2) Linux and macOS builds of the command schema and retrieval core; (3) gaze gestures such as blink confirmation; (4) multi-turn dialogue memory with quantized local LLMs; and (5) multi-center longitudinal clinical trials.

**Declarations**

**Funding:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Competing Interests:** The authors declare that they have no competing financial or non-financial interests.

**Ethics Approval and Consent:** The research was conducted using anonymized participant data. No personally identifiable information was collected, accessed, or disclosed during the study. Individual identities remained anonymous throughout the research process, ensuring that the privacy and confidentiality of all participants were protected. The findings are reported only in aggregated form, and no individual participant can be identified from the reported results.

**Availability of Data, Materials and Code: Availability of Data and Materials:** The synthetically generated template-based command corpus of 1,440 annotated file management utterances (partitioned into 1,008 train, 216 validation, and 216 test commands, alongside the 72-sample ASR test manifest), slot extraction benchmarks, and de-identified, aggregated participant interaction telemetry are available in the public project GitHub repository: https://github.com/GaurangPandey2912/voice-file-assistant.

**Authors' Contributions:** Gaurang Pandey: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing – Original Draft. Kirti Adhikari: Software, Validation, Data Curation, Visualization, Investigation, Writing – original draft. V. Rajalakshmi: Conceptualization, supervision, resources, project administration, formal analysis, writing—review & editing. All authors approved the final manuscript.

**References**

[1] Harada S, Wobbrock JO, Landay JA (2007) VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments. In: Proceedings of the 9th International ACM SIGACCESS

Conference on Computers and Accessibility (ASSETS '07), Tempe, AZ, USA, pp 27–34. https://doi.org/10.1145/1296843.1296850

[2] Khan AA, Weidner F, Rhee J, Abdrabou Y, Bianchi A, Velloso E, Gellersen H, Newn J (2026) Gaze and speech in multimodal human-computer interaction: A scoping review. In: Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26). ACM, Article 672, 24 pages. https://doi.org/10.1145/3772318.3791662

[3] Bhattacharya S, Basu A, Samanta D (2008) Computational modeling of user errors for the design of virtual scanning keyboards. IEEE Trans Neural Syst Rehabil Eng 16(4):400–409. https://doi.org/10.1109/TNSRE.2008.925073

[4] Dhanjal AS, Singh W (2024). A comprehensive survey on automatic speech recognition using neural networks. Multimed Tools Appl 83(8):23367–23412. https://doi.org/10.1007/s11042-023-16438-y

[5] Yerramreddy DR, Marasani J, Gowtham PSV, Harshit G, Anjali (2024) Speech recognition paradigms: A comparative evaluation of SpeechBrain, Whisper and Wav2Vec2 models. In: 2024 9th International Conference for Convergence in Technology (I2CT), Pune, India, pp 1–6. https://doi.org/10.1109/I2CT61223.2024.10544133

[6] Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech recognition using deep neural networks: A systematic review. IEEE Access 7:19143–19165. https://doi.org/10.1109/ACCESS.2019.2896880

[7] Xie T, Zhang D, Chen J, Li X, Zhao S, Cao R, Toh JH, Cheng Z, Shin D, Lei F, Liu Y, Xu Y, Zhou S, Savarese S, Xiong C, Zhong V, Yu T (2024) OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In: Advances in Neural Information Processing Systems 37 (NeurIPS 2024), pp 52040–52094. https://doi.org/10.52202/079017-1650

[8] Majaranta P, Bulling A (2014) Eye tracking and eye-based human-computer interaction. In: Fairclough S, Gilleade K (eds) Advances in Physiological Computing. Springer, London, pp. 39–65. https://doi.org/10.1007/978-1-4471-6392-3_3

[9] Mhawi DN, Oleiwi HW, Saeed NH, Al-Taie HL (2022) An efficient information retrieval system using enhanced inverted index technique. Bull Electr Eng Inform 11(4):2265–2272. https://doi.org/10.11591/eei.v11i4.3756

[10] Thakur N, Reimers N, Rücklé A, Srivastava A, Gurevych I (2021) BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Virtual, pp. 1–18. https://doi.org/10.48550/arXiv.2104.08663

[11] Reimers N, Gurevych I (2019) Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Inui K, Jiang J, Ng V, Wan X (eds) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp 3982–3992. https://doi.org/10.18653/v1/D19-1410

[12] Mitra B, Craswell N (2018) An introduction to neural information retrieval. Found Trends Inf Retr 13(1):1–126. https://doi.org/10.1561/1500000061

[13] Saritha K, Sridhar R, Rao KS (2021) Speech-driven real-time code editing assistant for programmers with motor disabilities. In: 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp 1–6

[14] Radford A, Kim JW, Xu T, Brockman G, McLeavey C, Sutskever I (2023) Robust speech recognition via large-scale weak supervision. In: Krause A, Brunskill E, Cho K, Engelhardt B, Sabato S, Scarlett J (eds) Proceedings of the 40th International Conference on Machine Learning, vol. 202. PMLR, Honolulu, Hawaii, pp. 28492–28518. https://doi.org/10.48550/arXiv.2212.04356

[15] Green JR, MacDonald RL, Jiang PP, Cattiau J, Heywood R, Cave R, Seaver K, Ladewig MA, Tobin J, Brenner MP, Nelson PC, Tomanek K (2021) Automatic speech recognition of disordered speech: Personalized models outperforming human listeners on short phrases. In: Interspeech 2021, Brno, Czech Republic, pp 4778–4782. https://doi.org/10.21437/Interspeech.2021-1384

[16] Rudzicz F, Namasivayam AK, Wolff T (2012) The TORGO database of acoustic and articulatory speech from speakers with dysarthria. Lang Resour Eval 46(4):523–541. https://doi.org/10.1007/s10579-011-9145-0

[17] Chandrakala CB, Bhardwaj R, Pujari C (2024) An intent recognition pipeline for conversational AI. Int J Inf Technol 16(2):731–743. https://doi.org/10.1007/s41870-023-01642-8

[18] Louvan S, Magnini B (2020) Recent neural methods on slot filling and intent detection for task-oriented dialogue systems: A survey. In: Scott D, Bel N, Zong C (eds) Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Barcelona, Spain, pp. 436–451. https://doi.org/10.18653/v1/2020.coling-main.39

[19] Weld H, Huang X, Long S, Poon J, Han SC (2022) A survey of joint intent detection and slot-filling models in natural language understanding. ACM Comput Surv 55(8):165:1–165:38. https://doi.org/10.1145/3547138

[20] Honnibal M, Montani I (2017) spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. Software available from https://spacy.io

[21] Sivasankari S, Gandhimathi R, Keerthana S, Mohanapriya R, Pavithra N (2020) File management using hand gesture and voice. Int Res J Eng Technol (IRJET) 7(4):5346–5349

[22] Needhi D, Shreya D, Jha DK, Jha A, Patil A (2022) Performance optimization of voice based file manager. In: 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, pp. 1–6. https://doi.org/10.1109/MysuruCon55714.2022.9972688

[23] Khan W, Daud A, Khan K, Muhammad S, Haq R (2023) Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and recent advances. Neurocomputing 559:126838. https://doi.org/10.1016/j.neucom.2023.126838

[24] Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J (2021) Deep learning-based text classification: A comprehensive review. ACM Comput Surv 54(3):62:1–62:40. https://doi.org/10.1145/3439726

[25] Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

Technologies, vol. 1. Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423

[26] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692 [cs.CL]. https://doi.org/10.48550/arXiv.1907.11692

[27] Sanh V, Debut L, Chaumond J, Wolf T (2019) DistilRoBERTa-base: Distilled RoBERTa transformer architecture and model weights. Hugging Face Model Hub. https://huggingface.co/distilbert/distilroberta-base; Sanh V et al. (2019) DistilBERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL]. https://doi.org/10.48550/arXiv.1910.01108

[28] Robertson S, Zaragoza H (2009) The probabilistic relevance framework: BM25 and beyond. Found Trends Inf Retr 3(4):333–389. https://doi.org/10.1561/1500000019

[29] Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M (2020) MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In: Larochelle H, Ranzato M, Hadsell R, DeFries MF, Lin HT (eds) Advances in Neural Information Processing Systems 33 (NeurIPS 2020). Curran Associates, Inc., Virtual, pp 5776–5788. https://doi.org/10.48550/arXiv.2002.10957

[30] Elghaish F, Chauhan JK, Matarneh S, Pour Rahimian F, Hosseini MR (2022) Artificial intelligence-based voice assistant for BIM data management. Automation in Construction 140:104320. https://doi.org/10.1016/j.autcon.2022.104320

[31] Scripka D (2023) openWakeWord: An open-source audio wake word detection toolkit. GitHub repository: https://github.com/dscripka/openWakeWord. Accessed 15 Jan 2024

[32] Silero Team (2021) Silero VAD: Pre-trained enterprise-grade voice activity detector. GitHub repository: https://github.com/snakers4/silero-vad

[33] Morris AC, Maier V, Green P (2004) From WER and RIL to MER and WIL: Improved evaluation measures for connected speech recognition. In: Interspeech 2004 - ICSLP, 8th International Conference on Spoken Language Processing. ISCA, Jeju Island, Korea, pp 2765–2768. https://doi.org/10.21437/Interspeech.2004-668

[34] Makhoul J, Kubala F, Schwartz R, Weischedel R (1999) Performance measures for information extraction. In: Proceedings of the DARPA Broadcast News Workshop, Herndon, VA, pp 249–252

[35] Walker MA, Litman DJ, Kamm CA, Abella A (1997) PARADISE: A framework for evaluating spoken dialogue agents. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL '97), Madrid, Spain, pp 271–280. https://doi.org/10.3115/976909.979652

[36] Rosenberg A, Ramabhadran B, Moreno P (2017) Towards task-oriented speech recognition. In: Interspeech 2017, Stockholm, Sweden, pp 2471–2475. https://doi.org/10.21437/Interspeech.2017-1082

[37] Craswell N (2009) Mean reciprocal rank. In: Liu L, Özsu MT (eds) Encyclopedia of Database Systems. Springer, Boston, MA, p 1703. https://doi.org/10.1007/978-0-387-39940-9_488

[38] Hart SG, Staveland LE (1988) Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. Adv Psychol 52:139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[39] Brooke J (1996) SUS: A 'quick and dirty' usability scale. In: Jordan PW, Thomas B, McClelland IL, Weerdmeester B (eds) Usability Evaluation in Industry. Taylor & Francis, London, pp. 189–194. https://doi.org/10.1201/9781498710411-35