Model Research

1 — Overview (what to include)

For each model in the research section include:

- **Model / Algorithm name**
- **Short description** (how it works)
- **Primary task(s)** (ASR, TTS, intent classification, routing, etc.)
- **Evaluation metrics** to report (e.g., WER, MOS, Accuracy, F1)
- **Typical performance range** (empirical range; run your own tests)
- **Training / test datasets** commonly **used** (for reproducibility)
- **Integration considerations** (latency, memory, API vs self-host)
- **Recommended baseline(s)** to compare against

2 — Models & Algorithms (by system component)

A. Speech → Text (ASR)

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Transformer-based ASR (e.g., Conformer, Transformer)	Uses self-attention + convolutional frontend to model long context in audio; trained end-to-end.	WER (Word Error Rate), Real-time factor (RTF), latency	WER varies by dataset: **low single digits to 10–20%** (clean vs noisy)
CTC (Connectionist Temporal Classification) models	Framewise outputs collapsed with CTC loss — good for streaming and monotonic alignment. Often paired with RNNs or convs.	WER, latency	Competitive on streaming; **WER** similar to other end-to-end ranges depending on data
RNN-Seq2Seq + Attention (Tacotron-style encoder/decoder for alignment)	Sequence-to-sequence mapping from audio features to tokens. Better for smaller datasets historically.	WER, latency	Older models; higher WER than modern transformers on large corpora
Self-supervised audio encoders (wav2vec2, HuBERT)	Pretrain on raw audio by contrastive/prediction losses; fine-tune on labeled ASR data → strong performance with less labeled data.	WER	Often **state of the art** on low-label regimes; WER improves markedly vs training from scratch

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Hybrid HMM-GMM / HMM-DNN	Classic pipeline: acoustic model + HMM decoder + pronunciation lexicon. Still useful for low-resource or deterministic pipelines.	WER	Solid baseline; usually beaten by SOTA end-to-end on large data but robust in constrained setups

Notes: report WER on standard splits (e.g., LibriSpeech test-clean/test-other) and on your in-domain audio. Also report latency (important for live chat/call).

B. Text → Speech (TTS)

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Tacotron 2 + Neural vocoder (WaveGlow / HiFi-GAN)	Sequence-to-sequence text→mel spectrogram then neural vocoder → waveform. Natural prosody.	MOS (Mean Opinion Score), MOS for naturalness, latency	MOS (naturalness) often **~3.5–4.5/5** for strong systems
FastSpeech / FastSpeech2	Non-autoregressive TTS for speed and stability. Needs vocoder.	MOS, real-time capability	Slightly more deterministic prosody; MOS similar to Tacotron when vocoder is good
Neural end-to-end TTS (VITS, Glow-TTS)	Single-model architectures that produce waveform directly or via latent flows — faster and often high quality.	MOS, latency	State-of-the-art quality and fast inference (good for real-time responses)

Notes: prefer models with streaming/low-latency support if replies must be immediate. Evaluate MOS with human raters or validated automatic proxies.

C. Dialogue / Chatbot Core (LLMs & Seq2Seq)

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Large Pretrained Transformers (GPT-style)	Autoregressive LLMs finetuned / prompted for dialogue. Strong fluency, context retention.	Task success, BLEU (not ideal), human eval, intent accuracy	Performance depends on model size & fine-tuning; in practice top LLMs give very high task accuracy on standard benchmarks (but evaluate on your tasks)
Encoder-	Good for controllable	Task accuracy,	Competitive for QA &

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Decoder Transformers (T5, BART)	response generation, retrieval-augmented setups, and fine-tuning on task datasets.	BLEU, ROUGE, human eval	summarization tasks
Retrieval-Augmented Generation (RAG)	Combines retrieval of documents/FAQ with generator to ground responses in knowledge.	Exact match / F1 on knowledge tasks, hallucination rate	Great for factual, up-to-date answers; reduces hallucinations if retrieval is good

Notes: measure task success (e.g., correct resolution rate), hallucination rate, response latency, and safety metrics (policy compliance).

D. Intent Detection & Slot Filling

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Fine-tuned BERT / RoBERTa / DistilBERT	Transformer encoders fine-tuned as classifiers for intent; slot filling via sequence tagging (BIO).	Accuracy, Precision, Recall, F1 (per intent & macro)	Intent classification accuracy typically **85–99%** depending on number of intents & data; F1 for slot filling **70–98%**
CRF + BiLSTM (for slot tagging)	BiLSTM feature extractor + CRF sequence tagger — strong for sequence labeling with limited compute.	F1 (slots)	Good baseline for structured slot tasks
Multi-task joint models (intent + slots in one model)	Joint losses improve consistency and end-to-end performance.	Intent accuracy, slot F1, end-to-end accuracy	Often improves real-world performance vs separate models

Notes: report per-intent confusion matrix and end-to-end dialog state accuracy.

E. Sentiment / Emotion Analysis

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Transformer classifiers (BERT, RoBERTa)	Fine-tuned on sentiment / emotion labels; can use text + prosody features for speech.	Accuracy, F1 (macro)	**~80–95%** depending on number of classes & domain

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Acoustic + Text multimodal models	Combine textual and audio prosody features for better emotion detection.	F1 (multimodal)	Typically improves recall for subtle emotions vs text only

Notes: provide confusion between “neutral” and other emotions; consider calibration for skewed classes.

F. Call Routing / Escalation Decisioning

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Gradient boosted trees (XGBoost / LightGBM)	Tabular features (intent, sentiment, history, confidence) → binary/multi class routing. Highly performant and explainable.	Accuracy, AUC, precision/recall for “escalate”	**AUC 0.85–0.98** in many corpora; accuracy depends on class balance
Neural nets / DNNs	Dense models for more complex feature interactions or sequence models for temporal history.	Accuracy, latency	Comparable to GBDT if enough data
Rule-based fallback + ML	Deterministic thresholds (low ASR confidence, profanity, timeouts) + ML for borderline cases.	Precision/Recall for escalations	Often used in production for safety & interpretability

Notes: optimize for **precision** on escalate (avoid false positives causing unnecessary agent involvement) or tune to business needs.

G. Response Ranking / Re-ranking

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Learning-to-Rank (LambdaMART, RankNet)	Rank candidate replies by relevance; trained on pairwise or listwise losses.	NDCG@k, MRR	Significant improvements over heuristic ranking; depends on training data
Cross-encoder re-rankers (BERT)	Compute relevance by joint encoding candidate+context (expensive but accurate).	NDCG, MRR	State-of-the-art for re-ranking

H. Recommendation Engine (FAQ / Next-best-action)

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
Collaborative Filtering / Matrix Factorization	Uses user-item interactions for personalized suggestions.	Precision@k, Recall@k, MAP	Good when user history exists; typical precision/recall strongly data dependent
LightFM / Neural CF	Hybrid models combining content & collaborative signals.	Precision@k, Recall@k	Better cold-start handling with content features

I. Call Quality / Audio Health Monitoring

Model / Algorithm	Summary	Metric(s) to report	Typical performance range
CNN or Transformer audio classifiers (e.g., wav2vec features + classifier)	Detect noise, dropouts, low SNR, packet loss.	Accuracy, F1	High detection rates with labeled examples; **>90%** for well-curated signals
Signal processing heuristics + ML fusion	Classic audio metrics (SNR, jitter) combined with ML for robust detection.	Precision/Recall	Useful to triage poor calls early

3 — Datasets (common choices & what to use)

- **ASR**: LibriSpeech (clean/other), Common Voice (multilingual), Switchboard (conversational), proprietary call recordings (must de-identify).
- **TTS**: LJSpeech, VCTK, proprietary speaker corpora for target voice.
- **Intent / Slots / Dialogue**: SNIPS, ATIS (small), MultiWOZ (multi-domain), plus curated in-domain intents.
- **Sentiment / Emotion**: IEMOCAP, SEMAINE, proprietary annotated call labels.
- **Routing / Escalation**: historical call logs labeled with escalate/no-escalate.
  Always evaluate on **in-domain** test sets — phone/VoIP audio, noisy environments, local languages/dialects.

4 — Evaluation metrics & reporting guidance

- **ASR**: WER (primary), CER (char error rate if logographic languages), Real-Time Factor, latency.
- **TTS**: MOS (human), MUSHRA or AB tests for voice naturalness; also synthesis latency.
- **Classification tasks** (intent, sentiment, routing): Accuracy, Precision/Recall, F1 (macro & per-class), confusion matrices.
- **Dialogue/LLM**: Task success rate (task completion), average turns to resolution, user satisfaction (human eval), hallucination rate.
- **Recommendation & Ranking**: Precision@k, Recall@k, NDCG@k, MRR.
- **Operational metrics**: computational cost, inference latency (ms), memory footprint, QPS, failure modes.

5 — Experimental setup (how to run fair comparisons)

1. **Train/Val/Test split** with stratification by intent/speaker/noise.
2. **Standardize pre-processing**: consistent feature extraction (sample rate, windowing, tokenization).
3. **Baseline models**: include 1 simple baseline per task (e.g., n-gram/CRF for slots, Random Forest for routing, classic HMM for ASR if used historically).
4. **Hyperparameter grid** documented for each run.
5. **Repeatability**: set random seeds, log checkpoints, and store evaluation scripts.
6. **Human evaluation** for TTS and end-to-end dialogue tasks.
7. **A/B testing** on live traffic for top candidates (monitor user satisfaction, escalate rates, agent load).

6 — Suggested baselines

- **ASR baseline**: wav2vec2 fine-tuned on your labeled audio + CTC decoder.
- **TTS baseline**: Tacotron2 + HiFi-GAN (or FastSpeech2 + HiFi-GAN) for quality/latency comparison.
- **Intent baseline**: logistic regression with TF-IDF + CRF for slots.
- **Routing baseline**: XGBoost on engineered features (intent, confidence, sentiment, recency).
- **Dialogue baseline**: retrieval system (FAQ retrieval) + small BART/T5 generator for answer polishing.
- **Recommendation baseline**: popularity + LightFM.

7 — Practical integration notes & tradeoffs

- **Latency vs Quality**: larger LLMs and cross-encoder rankers are more accurate but increase latency and cost. Use cascaded systems: fast lightweight model first, heavy model for ambiguous/higher-value queries.
- **On-premise vs API**: hosting models locally reduces latency and data exposure but increases infra burden. Cloud APIs speed up development. Consider hybrid (sensitive data on-prem, generic tasks via API).
- **Multilingual & accents**: fine-tune ASR and intent models on local accents and languages — generic models drop performance if not adapted.
- **Safety & Escalation**: always include explicit rules for safety/evasion/profanity and an explainable escalation path.
- **Data privacy**: de-identify call recordings; follow legal/regulatory requirements (GDPR, etc.).

8 — Example short write-up paragraph

Model research summary — We evaluated a suite of models across the speech and dialogue stack. For ASR we considered end-to-end transformer models (Conformer, wav2vec2) and CTC variants for streaming; typical WERs for state-of-the-art systems range from single digits on clean data to 10–20% in noisy real-world phone audio, so in-domain benchmarking is essential. For TTS we reviewed Tacotron2/ FastSpeech2 + neural vocoders (HiFi-GAN), which achieve high MOS scores for naturalness in modern pipelines. Dialogue capabilities were benchmarked between retrieval-based systems, encoder–decoder models (BART/T5), and large autoregressive LLMs; retrieval-augmented generation reduces factual errors in knowledge-grounded responses. Key ML subsystems (intent detection, sentiment, routing) are best implemented by fine-tuned transformer encoders or gradient-boosted trees depending on data volume; classification accuracy commonly ranges from mid-80s to high-90s percent with sufficient labeled data. All performance figures are dataset and domain dependent; we recommend follow-up benchmarks using our in-domain call recordings and user transcripts, plus human evaluation for any naturalness or satisfaction metrics.

9 — Appendix:

- **ASR**: WER (lower = better)
- **TTS**: MOS (higher = better)
- **Intent/Slot**: Accuracy, F1 (macro & per class)
- **Routing**: AUC, Precision@Recall thresholding
- **Ranking**: NDCG@k, MRR
- **Operational**: Latency (ms), RTF, memory, cost per 1k requests