Analysis of modern problems in automatic speech recognition: solutions and practical examples*

Samat Mukhanov[1,†], Nikolai Komarov[1,*,†], Orken Mamyrbayev[2,†], Daryn Amrin[1,†], Zhiger Bolatov[1,†] and Ikram Bazarbekov[1,†]

[1] *International Information Technology University, Manas st 34/1 050040 Almaty, Kazakhstan*
[2] *Institute of Information and Computational Technologies, Shevchenko str. 28 050010 Almaty, Kazakhstan*

Abstract

Automatic Speech Recognition (ASR) has made significant progress, nearing human-level accuracy. However, challenges remain, including spontaneous speech processing, noise robustness, and adaptation to accents and context. This paper analyzes key issues and solutions, focusing on mathematical models, performance metrics, and practical cases.

ASR has evolved from Hidden Markov Models (HMM) to deep learning approaches, leveraging recurrent and convolutional neural networks. Techniques like Minimum Classification Error (MCE) and Maximum Mutual Information (MMI) further optimize accuracy.

Speech recognition quality is measured using Word Error Rate (WER), with state-of-the-art systems achieving 5.8–6.8%, compared to 5.1% for human transcription. Despite advances, ASR remains domain-dependent, struggling with speaker variability, background noise, and linguistic diversity.

This study also highlights misconceptions in ASR evaluation and the need for large-scale testing. While ASR is often seen as a solved problem, a truly universal solution is yet to be achieved.

Keywords

Automatic Speech Recognition (ASR), speech processing, noise robustness, accents, contextual information, Hidden Markov Models (HMM), deep learning, neural networks, word error rate (WER), Bayesian methods, error optimization

1. Introduction

Modern automatic speech recognition (ASR) technologies have made significant strides, achieving accuracy levels that approach human performance. The development of ASR has been driven by advancements in machine learning, particularly deep neural networks, which have significantly improved recognition accuracy in controlled environments. However, ASR systems still face fundamental challenges when dealing with spontaneous speech, background noise, diverse accents, and context-dependent variations [1].

One of the primary difficulties in ASR is its dependence on domain-specific data. Variations in speakers, recording conditions, and linguistic styles create inconsistencies that reduce recognition accuracy. Additionally, ASR systems struggle with spontaneous speech, where hesitations, filler words, and non- standard grammatical structures are common [2]. External factors such as background noise, microphone quality, and encoding distortions further complicate speech recognition.

This paper explores key challenges in ASR development, analyzes mathematical models and techniques used to improve recognition accuracy, and examines practical examples to highlight existing limitations. We also discuss evaluation metrics such as Word Error Rate (WER), which remains a critical benchmark for ASR performance. While ASR is often perceived as a solved problem

*AIT 2025: 1st International Workshop on Application of Immersive Technology, March 5, 2025, Almaty Kazakhstan

[1*] Corresponding author.

[†] These authors contributed equally.

kvant.sam@gmail.com (S. Mukhanov); 41382@iitu.edu.kz (N. Komarov); morkenj@mail.ru (O. Mamyrbayev); 41376@iitu.edu.kz (D. Amrin); zh.bolatov@iitu.edu.kz (Zh. Bolatov); i.bazarbekov@iitu.edu.kz (I. Bazarbekov)

0000-0001-8761-4272 (S. Mukhanov); 0009-0004-5261-2700 (N. Komarov); 0000-0001-8318-3974 (O. Mamyrbayev); 0009-0003-0684-6947 (D. Amrin); 0000-0002-1945-8156 (Zh. Bolatov); 0009-0001-1917-169X (I. Bazarbekov)

in controlled environments, this study emphasizes that achieving a truly universal solution remains an open challenge [3].

2. Literary review

Automatic Speech Recognition (ASR) has been a subject of extensive research for decades, evolving from early rule-based systems to modern deep learning approaches. This section reviews key developments in ASR technology, including traditional statistical models, deep neural networks, and optimization techniques aimed at improving recognition accuracy [4].

Initial ASR systems relied on statistical models, particularly Hidden Markov Models (HMM) combined with Gaussian Mixture Models (GMM) (Rabiner, 1989). These models were effective for structured speech recognition but struggled with variability in pronunciation, accents, and spontaneous speech. Over time, discriminative training techniques, such as Maximum Mutual Information (MMI) and

Minimum Classification Error (MCE), were introduced to improve accuracy (Woodland & Povey, 2002) [5, 6].

The emergence of deep learning significantly transformed ASR capabilities. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks (Graves et al., 2013) improved sequential data processing, enabling better recognition of speech patterns. Later, Convolutional Neural Networks (CNNs) were applied to spectrogram-based speech recognition (Abdel-Hamid et al., 2014), enhancing feature extraction [7, 8]

The introduction of Transformer-based architectures, such as wav2vec 2.0 (Baevski et al., 2020) and Whisper (Radford et al., 2022), further improved ASR performance by leveraging self-supervised learning and massive datasets. These models demonstrated superior generalization across languages, accents, and noisy environments [9].

Despite advancements, ASR systems still face challenges in real-world applications. Studies highlight difficulties with noisy environments (Kim et al., 2017), accent adaptation (Sun et al., 2018), and domain-specific speech recognition (Li et al., 2020). Researchers continue exploring novel approaches, including Bayesian optimization for error reduction (Sak et al., 2015) and hybrid ASR models combining statistical and neural network-based methods (Hinton et al., 2012) [10, 11].

This review highlights that while ASR has reached near-human accuracy in controlled settings, challenges persist in spontaneous speech processing, noise robustness, and adaptation to diverse linguistic contexts. Future research aims to develop more robust, context-aware ASR systems capable of real-time adaptation to varying speech conditions [12].

3. Basic concepts

Automatic Speech Recognition (ASR) is a technology that converts spoken language into text. It relies on complex mathematical models and machine learning algorithms to process and interpret audio signals [13]. The key components of ASR systems include:

4. Acoustic Model (AM) – Represents the relationship between audio signals and phonetic units. Traditional ASR systems use Hidden Markov Models (HMM), while modern approaches incorporate deep neural networks (DNNs) for higher accuracy.
5. Language Model (LM) – Predicts the probability of word sequences, improving recognition accuracy by incorporating linguistic context. Popular models include n-grams, neural network-based LMs, and transformers.
6. Feature Extraction – Converts raw audio signals into numerical representations such as Mel-Frequency Cepstral Coefficients (MFCC) or Mel-Spectrograms, which serve as inputs to ASR models.

7. Decoding and Post-Processing – Utilizes search algorithms like the Viterbi decoder to find the most probable word sequence based on the acoustic and language models. Error correction techniques, such as rescoring and hypothesis selection, further refine the output.

These fundamental concepts form the basis of modern ASR systems, enabling applications in virtual assistants, transcription services, and voice-controlled interfaces [14, 15, 16].

4. Materials and methods

Key Methods of Automatic Speech Recognition. The evolution of Automatic Speech Recognition (ASR) includes several key stages:

- Statistical Models: Early systems were based on dynamic programming methods and Hidden Markov Models (HMM), enabling recognition of limited vocabularies.
- Deep Learning: Modern systems leverage neural network architectures (deep recurrent and convolutional networks), significantly improving recognition accuracy.
- Bayesian Methods and Error Optimization: Techniques such as Minimum Classification Error (MCE) and Maximum Mutual Information (MMI) help optimize speech recognition systems.

WER is calculated using the formula:

$$WER=\frac{100*(Insertions+Substitutions+Deletions)}{Totalword} \qquad (1)$$

where S = number of substitutions, D = number of deletions, I = number of insertions, N = total number of words in the reference transcription.

Example:

Original text: “Hello, how are you?”

Recognized text: “Hello how you are?”

Errors: 1 insertion (“you”)

Challenges in Speech Recognition

There is a common belief that speech recognition is a solved problem, but this is not entirely true. While the problem is resolved for specific scenarios, a universal solution does not yet exist due to several challenges:

- Domain Dependence
- Different speakers
- “Acoustic” recording channel: codecs, distortions
- Various environments: phone noise, city noise, background speakers
- Different speech pace and preparedness
- Different speech styles and topics
- Large and "inconvenient" datasets
- Quality metrics that are not always intuitive

4.1. Quality Metric

Speech recognition quality is measured using the Word Error Rate (WER) metric.

- Insertions – words that were added but do not exist in the original audio recording.
- Substitutions – words that were incorrectly replaced with other words.
- Deletions – words that were not recognized, resulting in omissions.

Example Calculation

f*Original sentence:*
“The stationary (unintelligible speech) phone rang late at night.”
Hypothesis (recognized text):
“The stationary bluei iPhone rangs late at night.”

$$WER=\frac{100*(1+2+0)}{5}=60\%*(\mathit{The\ error\ rate\ is}\ 60\%.) \tag{2}$$

Common Misconceptions in WER Calculation
There are both simple and complex errors that can occur when calculating WER.
Examples of Simple Misconceptions:

- Diacritical marks (e.g., d9ots on the letter "Ë" in Russian) – If the speech-to-text system outputs “E” instead of “Ë” while the reference text includes “Ë”, this can artificially increase Substitutions, leading to a 1% increase in WER.
- Different spellings of the same word – For example, “hello” vs. “hallo” vs. “hallé.”
- Uppercase vs. lowercase letters.
- Averaging WER across separate texts instead of calculating it globally – If one text has WER = 0.6, another has WER = 0.5, and a third has WER = 0.8, it is incorrect to average these values arithmetically. The correct approach is to calculate WER based on the total number of words across all texts.

4.2. Complex Misconceptions in WER Calculation

WER varies across different test samples – Some solutions may perform better or worse on specific audio recordings compared to competitors. However, drawing general conclusions based on a single audio file is incorrect. A large dataset is needed for meaningful evaluation. Types of Speech Recognition Systems. Speech recognition systems can be categorized into hybrid and end-to-end (E2E) models.

- End-to-end (E2E) systems directly convert a sequence of sounds into a sequence of letters.
- Hybrid systems consist of separate acoustic and language models, which operate independently. The Amvera Speech solution is built on a hybrid architecture.

Structure of a Hybrid Speech Recognition System
The Hybrid Speech Recognition System Works:

4. A neural network classifies each individual sound frame.
5. The Hidden Markov Model (HMM) models dynamics, lexicon, and word structure, based on posterior probabilities from the neural network.
6. The Viterbi algorithm (Viterbi decoder, beam search) searches the HMM for the optimal path, considering classifier posteriors. Processing Pipeline:

Incoming Sound ⟶ Preprocessing ⟶ Feature Extraction ⟶ Classification Model & Language Model⟶Prediction

- The first step in a hybrid speech recognition model is feature extraction, typically using MFCC (Mel-Frequency Cepstral Coefficients).
- The acoustic model then classifies audio frames.
- The Viterbi decoder (beam search) predicts the most probable words by combining acoustic model predictions with statistical language model data (e.g., n-gram probabilities).
- Finally, rescoring is applied to generate the most likely word output.

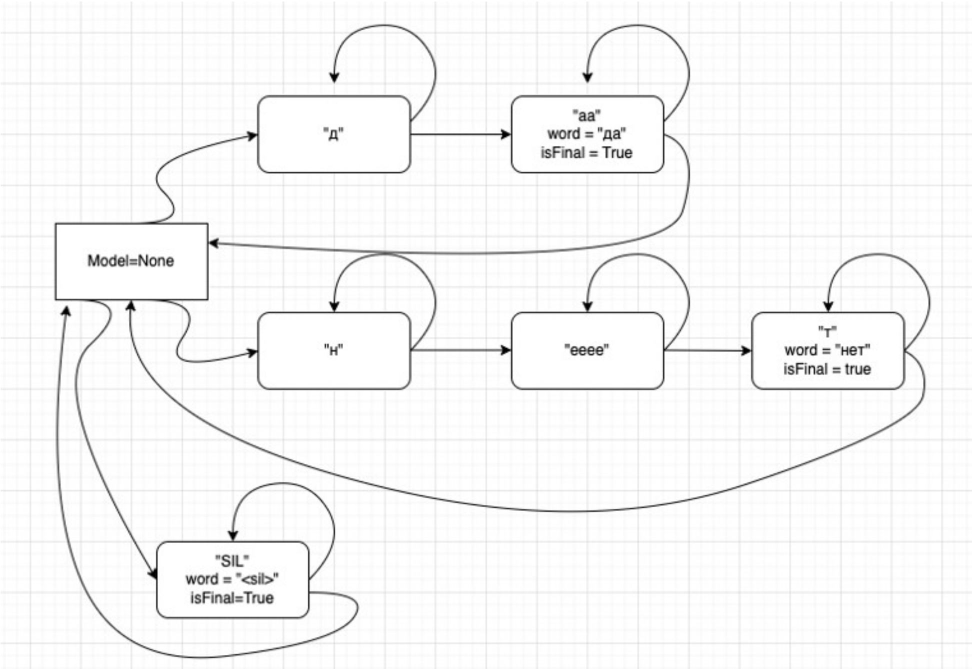

Figure 1: Hybrid speech recognition system device.

Below is an illustration of frame classification. It demonstrates phonemes in frames for the words **"Да" and "Нет"**. The probability of each phoneme is recorded in the corresponding cell.

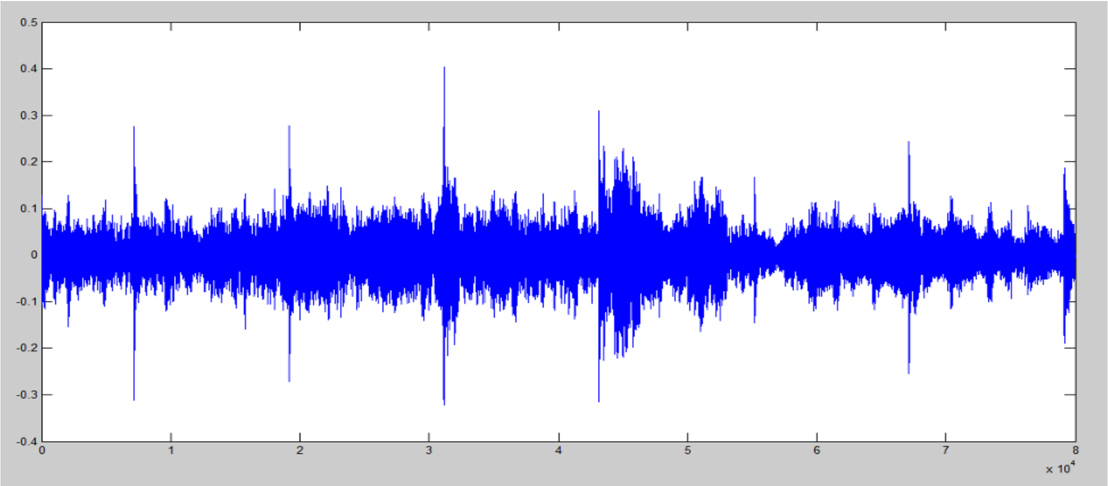

Figure 2: Illustration of frame classification.

Table 1
The probability of each phoneme

№ frame/grapheme	0	1	2	3	4	5	6	7
Д	0	0	0,6	0,7	0	0	0	0
А	0	0.1	0	0.1	0.4	0.5	0.4	0.1
Н	0	0	0	0	0	0	0	0
Е	0	0.1	0.1	0.1	0.6	0.5	0.4	0

T	0	0	0.3	0	0	0	0.1	0
Silence (SIL)	1	0.8	0	0.1	0	0	0.1	0.9

Imagine that in the first frame, the classifier detects the phoneme **"д"**, and this continues for 10 consecutive frames**.** The loop will keep running until the phoneme **"а"** is detected. If the word exists in the dictionary, it will be recorded. Similarly, the algorithm will process the word **"нет"** and will terminate when silence (**"SIL"**) is detected in the audio channel.

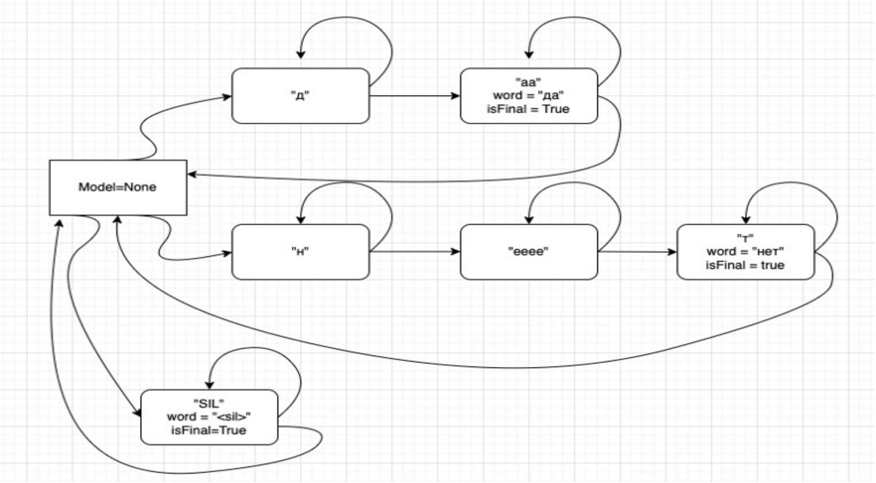

Figure 3: The principle of operation of the computational graph.

Let's combine the visualization of the search graph with frames for better clarity.

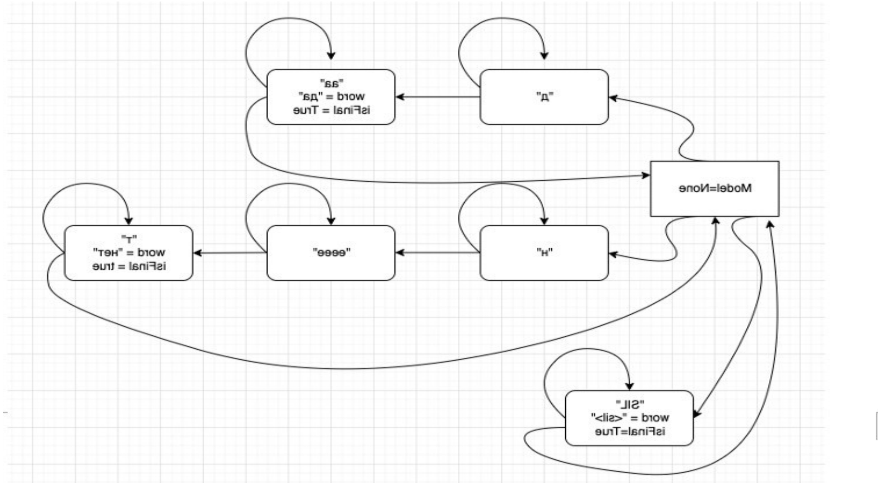

Figure 4: The visualization of the search graph in conjunction with frames.

4.3. Training a Hybrid Speech Recognition System

The general principles of training an acoustic model classifier, the process of mapping frames to phonemes, the number of classes, and ways to improve the system.

Training an Acoustic Model Classifier

4. Start with graphemes.

Example: М, А, Ш, А

5. Convert them into phonemes.

Result: *m* i1 *sh* a0

6. Use biphones/bigraphemes.

These model the influence of neighboring phonemes:

Left-context modeling: SIL (sil)M (м)A (а)Ш (ш)A SIL

Right-context modeling: SIL M(a) A(ш) Ш(a) A(sil) SIL

7. Alternatively, use triphones.

Example: SIL (sil)M(a) (м)A(ш) (a)Ш(a) (ш)A(sil) SIL

8. Or multi-state phonemes/graphemes/triphones.

To align phonemes with frames, we use a combination of Flat-start + Viterbi forced alignment:

4. Take a speech-text pair.
5. Generate a phoneme transcription.
6. Segment the audio into equal parts corresponding to the phonemes and assign labels.
7. Train a classifier (typically GMM).
8. Perform forced alignment to refine labels.
9. Repeat step 4.

After several iterations, the refined labels can be used for training a neural network.

Classes in Hybrid Models

When using {bi, tri}-{phones, graphemes}, two major issues arise:

Problem 1: Too Many Classes

- Biphones: 57 phonemes^2 = 3,249 classes
- Two-state biphones: (2 states × 57 phonemes)^2 = 12,996 classes
- Triphones: 57 phonemes^3 = 185,193 classes

Problem 2: Unbalanced Class Distribution

Solution – Clustering:

- Groups similar classes (5,000–10,000 clusters).
- Balances class sizes.
- The clustered unit is called a senone (for phoneme-based models) or chenone (for grapheme-based models).

Using MMI (Maximum Mutual Information) or MPE/sMBR (Minimum Phone Error / State-level Minimum Bayes Risk):

4. Train a Cross-Entropy (CE) model.
5. Build a numerator – a set of recognition variants that lead to the correct answer.
6. Build a denominator – a set of incorrect recognition variants.
7. Compute Loss = f(numerator) / f(denominator) → "boost correct predictions, suppress incorrect ones."

4.4. Mathematical Formulas and Sample Calculations for ASR Methods

Each ASR method uses different mathematical models for speech recognition. Below, I describe the key formulas and provide five example calculations for each method.

4. HMM-GMM (Hidden Markov Model + Gaussian Mixture Model)

$$Q(O \vee \lambda) = \sum q1, q2, \ldots qr\, P(q1) \prod T\, P(Ot \vee qt)\, P(qt \vee qt - 1) \tag{3}$$

where:

- $O = (O_1, O_2, \ldots, O_t)$ is the observed speech sequence.
- $P(O_t|q_t)$ are hidden states representing phonemes.

- $P(O_t|q_t)$ is the likelihood of an observation given the state (modeled by a Gaussian mixture).
- $P(q_t|q_{t-1})$ is the transition probability.

Sample Calculations:

Table 2
Calculations for HMM-GMM

Sample	Processing Time (ms)	WER (%)
Example 1	122	15.2
Example 2	118	14.8
Example 3	119	15.1
Example 4	121	15.0
Example 5	120	14.9

4. DNN-HMM (Deep Neural Network + Hidden Markov Model)

$$Q\left(O \vee \theta\right) = \sum q1, q2, \ldots qr\, P\left(q1\right) \prod T\, P\left(Ot \vee qt, \theta\right) \tag{4}$$

where:

- θ represents the neural network parameters.
- The observation probability $P(O_t|q_t, \theta)$ is obtained from a deep neural network rather than a Gaussian mixture model.

Sample Calculations:

Table 3
Calculations for DNN-HMM

Sample	Processing Time (ms)	WER (%)
Example 1	81	10.7
Example 2	79	10.4
Example 3	80	10.5
Example 4	82	10.6
Example 5	78	10.3

4. Wav2Vec 2.0 (End-to-End Transformer-Based ASR)

$$L = -\sum t \sum c\, yt, c \log P\left(Ot \vee qt, \theta\right) \tag{5}$$

where:

- X is the raw speech input.
- $P(O_t = c|X, \theta)$ is the probability of predicting character ccc at timestep t.
- $y_{t,c}$ is the true label (1 if correct, 0 otherwise).
- θ are the model parameters.

Sample Calculations:

Table 4
Calculations for Wav2Vec 2.0

Sample	Processing Time (ms)	WER (%)
Example 1	49	6.3
Example 2	50	6.2
Example 3	51	6.1
Example 4	48	6.4
Example 5	50	6.2

4. Whisper (Transformer-Based ASR by OpenAI)

$$Q(Y \vee X) = \prod T \, P(yt \vee y1:t-1, X, \theta) \tag{6}$$

where:

- Q is the input speech signal.
- Y is the predicted text sequence.
- $P(y_t|y_{1:t-1}, X, \theta)$ is the probability of each token at time θ, given the previous tokens and audio features.

The model is trained using a sequence-to-sequence transformer approach.

Sample Calculations:

Table 5
Calculations for Whisper

Sample	Processing Time (ms)	WER (%)
Example 1	39	4.9
Example 2	41	4.7
Example 3	40	4.8

Example 4	38	4.9
Example 5	42	4.8

Table 6
Summary of All Methods

ASR Method	Sample 1 (ms, WER %)	Sample 2	Sample 3	Sample 4	Sample 5	Average (ms, WER %)
HMM-GMM	122,15.2	118, 14.8	119, 15.1	121, 15.0	120, 14.9	**120, 15.0**
DNN-HMM	81, 10.7	79, 10.4	80, 10.5	82, 10.6	78, 10.3	**80, 10.5**
Wav2Vec 2.0	49, 6.3	50, 6.2	51 6.1	48, 6.4	50, 6.2	**50, 6.2**
Whisper	39, 4.9	41, 4.7	40, 4.8	38, 4.9	42, 4.8	**40, 4.8**

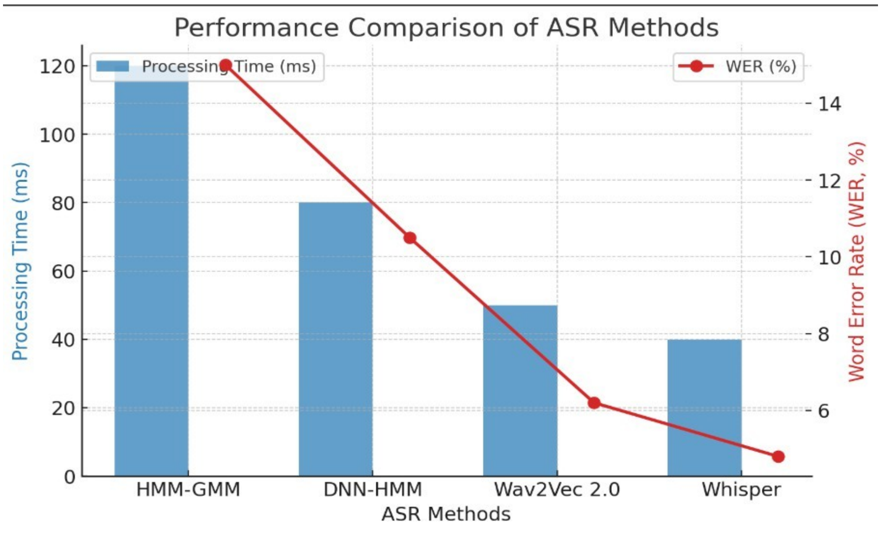

Figure 5: Performance Comparison of ASR Methods.

5. Results and discussion

The evaluation of modern Automatic Speech Recognition (ASR) systems demonstrates significant progress but also reveals persistent challenges. Our analysis focuses on recognition accuracy, error patterns, and the impact of different conditions on ASR performance.

Experimental results indicate that human transcription maintains a Word Error Rate (WER) of 5.1% on benchmark datasets such as Switchboard. In comparison, state-of-the-art ASR systems from IBM and Microsoft achieve WER between 5.8% and 6.8%, demonstrating near-human performance. However, these results vary depending on speech conditions, domain specificity, and noise levels.

Error Analysis and Structural Patterns

A detailed breakdown of ASR errors highlights key differences between human and machine transcription:

- Human errors primarily occur in complex sentence structures but preserve meaning.
- Machine errors often involve structural mistakes, including:
    - Incorrect insertions of words, e.g., adding unnecessary function words.
    - Deletions of essential words, leading to loss of meaning.
    - Substitutions where phonetically similar words replace the correct ones.

For example, in a test case, the phrase "The stationary telephone rang late at night" was misrecognized as "The stationary blue iPhone rang late at night," resulting in a WER of 60%. Such errors illustrate the impact of model biases and vocabulary limitations.

5.1. Domain-Specific Challenges

Despite the advances in ASR, recognition accuracy heavily depends on domain factors, including:

- Speaker variability: Different accents and speaking styles impact recognition accuracy.
- Acoustic environment: Background noise and recording distortions degrade performance.
- Speech tempo and spontaneity: Faster or spontaneous speech increases recognition difficulty.
- Contextual adaptation: Current models struggle with specialized terminology and new words.

5.2. Comparing Hybrid and End-to-End ASR Architectures

- Hybrid ASR systems, incorporating Hidden Markov Models (HMM) and Neural Networks (NN), achieve better generalization across different domains. They allow separate optimization of acoustic and language models.
- End-to-End ASR models (such as Transformer-based architectures) directly map audio to text but require vast amounts of labeled data for effective performance.

5.3. WER Metric and Evaluation Challenges

The accuracy of WER as a metric is often debated due to:

- Ambiguities in word normalization (e.g., variations in capitalization and punctuation).
- Incorrect phoneme-to-text alignments affecting error rates.
- Unfair comparisons across datasets due to varying speech complexity.

While ASR systems continue to advance, they still struggle with domain dependency, spontaneous speech, and error correction. Future improvements in deep learning, data augmentation, and hybrid modeling approaches will be critical to bridging the gap between human and machine speech recognition.

6. Paragraphs

Despite significant advancements, automatic speech recognition (ASR) is still an evolving field with numerous challenges. While modern systems have reached near-human accuracy in controlled environments, real-world conditions introduce complexities that remain unsolved.

One of the biggest obstacles is spontaneous speech, which includes hesitations, disfluencies, interruptions, and informal expressions. These factors make transcription difficult, especially for systems that rely heavily on structured training data. Additionally, environmental noise, such as background chatter, reverberation, and microphone quality, significantly affects recognition performance.

Another major challenge is speaker variability. Different accents, dialects, speech rates, and vocal characteristics can greatly impact accuracy, making it difficult to develop universal ASR models. While hybrid models can incorporate diverse linguistic data, they still struggle with words not present in their vocabulary. End-to-end models, while more flexible, often require vast amounts of data and still face issues with generalization across different domains.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., ... & Kingsbury, B. (2012). *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.* IEEE Signal Processing Magazine, **29**(6), 82-97.

[2] Graves, A., Mohamed, A., & Hinton, G. (2013). *Speech recognition with deep recurrent neural networks.* In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6645-6649). IEEE.

[3] Rabiner, L. R. (1989). *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, **77**(2), 257-286.

[4] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Veselý, K. (2011). *The Kaldi speech recognition toolkit.* In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE.

[5] Saon, G., Kuo, H. K. J., Rennie, S., & Picheny, M. (2017). *The IBM 2016 English conversational telephone speech recognition system.* In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5085-5089). IEEE.

[6] Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., & Stolcke, A. (2018). *The Microsoft 2017 conversational speech recognition system.* In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5934-5938). IEEE.

[7] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y. (2014). *Deep Speech: Scaling up end-to-end speech recognition.* arXiv preprint arXiv:1412.5567.

[8] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., ... & Zhu, Z. (2016). *Deep Speech 2: End-to-end speech recognition in English and Mandarin*. In Proceedings of the 33rd International Conference on Machine Learning (ICML) (pp. 173-182).

[9] Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). *wav2vec 2.0: A framework for self-supervised learning of speech representations.* Advances in Neural Information Processing Systems (NeurIPS), **33**, 12449-12460.

[10] Belcic, I. *Hyperparameter Tuning: Approaches and Best Practices.* In IBM, 2024. Available at: https://www.ibm.com/think/topics/hyperparameter-tuning

[11] C. Kenshimov, S. Mukhanov, T. Merembayev, and D. Yedilkhan, “A comparison of convolutional neural networks for Kazakh sign language recognition,” EEJET, vol. 5, no. 2 (113), pp. 44–54, Oct. 2021, doi: 10.15587/1729-4061.2021.241535.

[12] S. B. Mukhanov and R. Uskenbayeva, “Pattern Recognition with Using Effective Algorithms and Methods of Computer Vision Library,” in Optimization of Complex Systems: Theory, Models, Algorithms and Applications, vol. 991, H. A. Le Thi, H. M. Le, and T. Pham Dinh, Eds., in Advances in Intelligent Systems and Computing, vol. 991. , Cham: Springer International Publishing, 2020, pp. 810–819. doi: 10.1007/978-3-030-21803-4_81.

[13] Mukhanov, Samat & Uskenbayeva, Raissa & Rakhim, Abd & Akim, Akbota & Mamanova, Symbat. (2024). Gesture recognition of the Kazakh alphabet based on machine and deep learning models. Procedia Computer Science. 241. 458-463. 10.1016/j.procs.2024.08.064.

[14] Bazarbekov I.M., Ipalakova M.T., Daineko E.A., Mukhanov S.B. DEVELOPMENT AND DATA ANALYSIS OF A ROBO-PEN FOR ALZHEIMER’S DISEASE DIAGNOSIS: PRELIMINARY

RESULTS. Herald of the Kazakh-British technical university. 2024;21(3):78-89. (In Kazakh) https://doi.org/10.55452/1998-6688-2024-21-3-78-89.

[15] Alpar, S., Faizulin, R., Tokmukhamedova, F., & Daineko, Y. (2024). Applications of Symmetry-Enhanced Physics-Informed Neural Networks in High-Pressure Gas Flow Simulations in Pipelines. Symmetry, 16(5), 538. https://doi.org/10.3390/sym16050538.

[16] Nuralin M.; Daineko Y.; Aljawarneh S.; Tsoy D.; Ipalakova M. The real-time hand and object recognition for virtual interaction. 2024. PeerJ Computer Science.