# PARAPHASE GENERATION

**Onkar Katkamwar** - 202201040098
**Vidhisha Deshmukh** - 202201040087
**Prajwal Ganar** - 202201040034

**Guide** - Mrs.Diptee Ghusse

# Index

- Introduction
- Problem Statement
- Research Paper Summary
- Architecture Overview
- Without Attention (LSTM/GRU)
- With Attention (Luong/Bahdanau)
- Self-Attention (Transformer-based)
- Performance Comparison
- Evaluation Metrics Comparison
- Conclusion
- References

# Introduction

- Paraphrase Generation is the task of generating a sentence that conveys the same meaning as a given sentence but using different words or structure.
- It is a text-to-text generation task and a key challenge in NLP.

## Applications:

- Question-answering systems
- Chatbots
- Semantic search
- Data augmentation for NLP models

# Research Paper

## Generating Sentences from a Continuous Space

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2016). Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349. https://arxiv.org/pdf/1511.06349

Aim:
To build a generative language model that learns overall sentence characteristics like style, topic, and syntax using a continuous latent space, and to show it can produce coherent and diverse sentences by sampling from that space.

Objectives
To propose a Variational Autoencoder (VAE) architecture for text, particularly sentences.
To investigate whether incorporating a global latent variable helps in modeling holistic sentence properties.
To explore the smoothness and interpretability of the latent space by interpolating between sentence encodings.
To evaluate the model's performance in tasks like sentence reconstruction and word imputation.

# Methodology

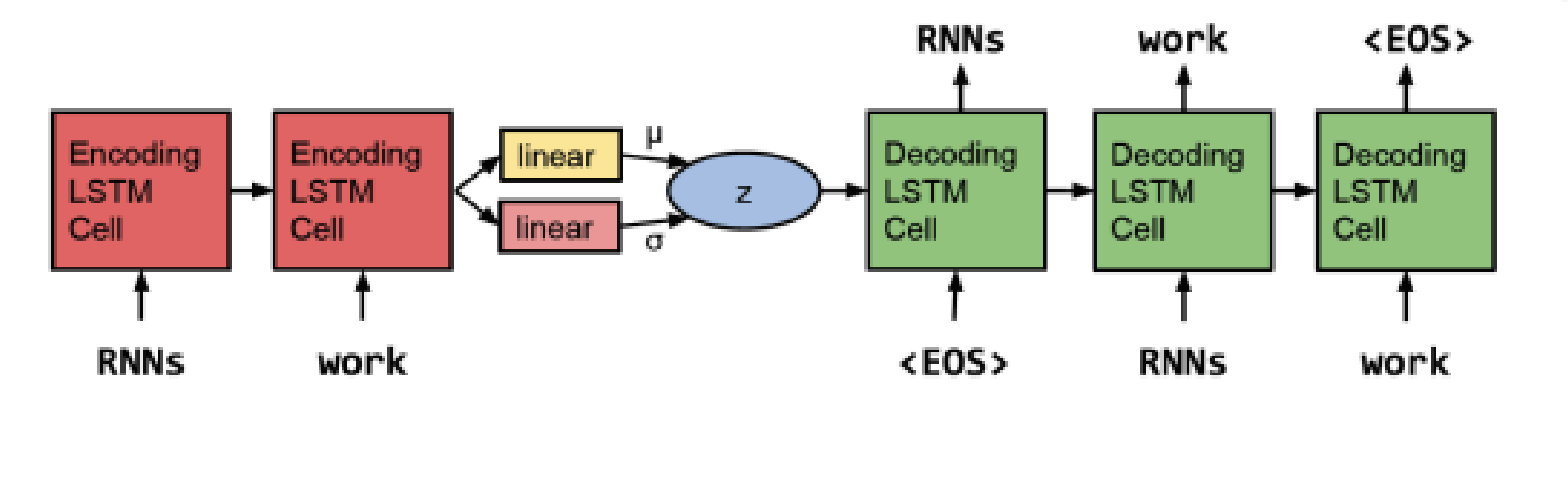
An encoder-decoder framework based on LSTM networks.

The encoder maps an input sentence to a latent Gaussian distribution ($q(z|x)$).

The decoder is an RNN conditioned on a sampled z from this distribution, reconstructing the sentence.

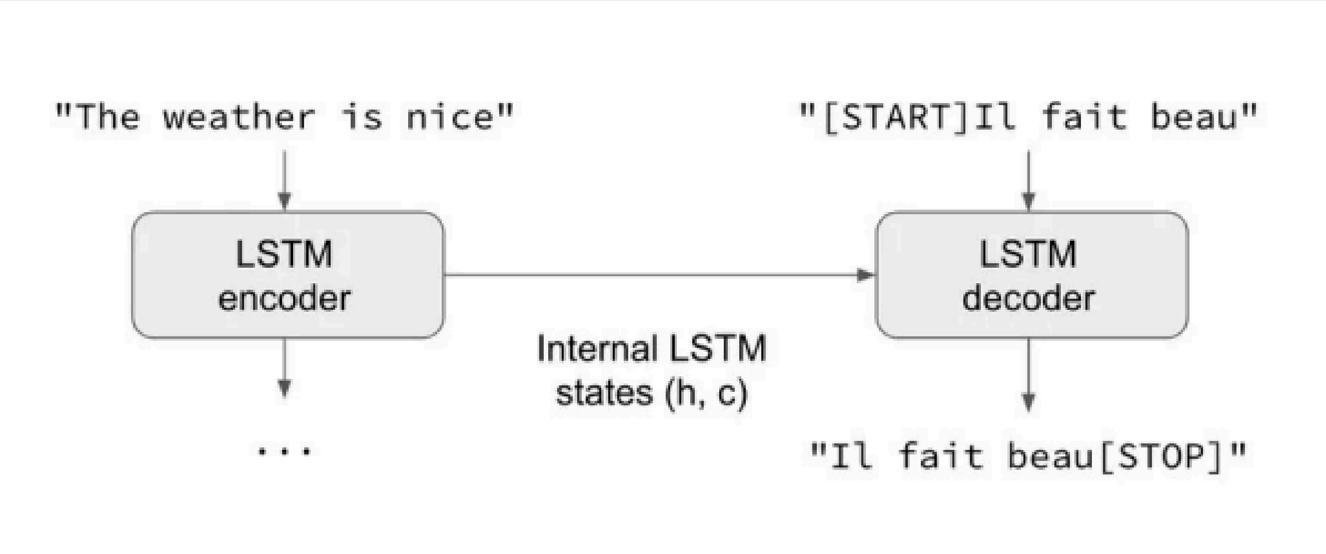
Trained using the reparameterization trick and stochastic gradient descent.

Encourages posterior distributions to be close to a standard Gaussian prior.



# Problem Statement

- Goal: Generate syntactically different but semantically similar sentences.
- Challenges:
  - Capturing long-range dependencies
  - Preserving meaning
  - Generating grammatically correct output
- Traditional methods (e.g., rule-based) fail in diverse contexts.
- Neural sequence-to-sequence (Seq2Seq) models are now standard.

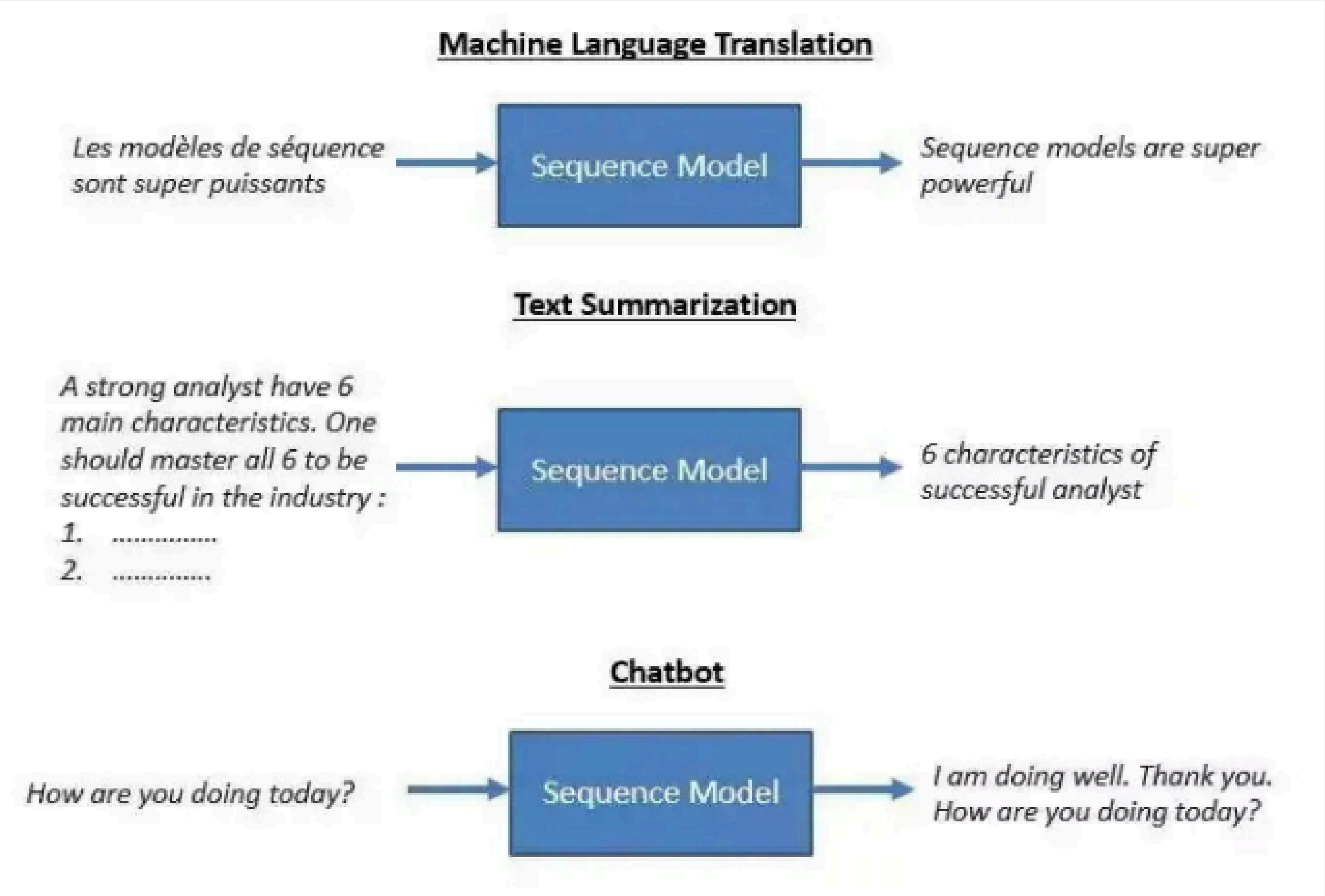


# Architecture Overview

All models are based on Seq2Seq architecture:

- **Encoder**: processes input sentence into context representation.
- **Decoder**: generates output sentence token by token.

- Variants differ in how they use context:
  - **No Attention**: fixed vector
  - **With Attention**: dynamic context via alignment
  - **Self-Attention**: token-to-token interactions at all levels



# Without Attention (LSTM/GRU)

**Simple LSTM-based encoder-decoder:**
Encoder reads entire sentence and compresses it into one context vector.
Decoder uses only this vector to generate output.

**Drawbacks:**
Performance drops for longer sequences. No access to individual words during decoding.
No interpretability: We can’t see what the model is focusing on.

| Component | Details |
|---|---|
| **Encoder** | Single LSTM, returns final state (summary vector) |
| **Decoder** | LSTM initialized with encoder state, predicts one token at a time |
| **No Attention** | Decoder can't refer to specific encoder tokens — only the final state |
| **Loss** | Sparse categorical crossentropy |
| **Optimizer** | Adam — fast, adaptive learning rates |

# With Attention (Luong/Bahdanau)

**Bahdanau Attention:**
Computes alignment scores between decoder state and encoder outputs.

**Soft attention:** all encoder states contribute, weighted by relevance.

**Luong Attention:** Uses dot product for alignment and can be global or local.

**Benefits:** Handles longer sentences better.
Adds interpretability (attention heatmaps).
Improves output fluency and accuracy.

# Self-Attention (Transformer-based)

**Self-Attention allows each word to attend to all others:**

- Captures dependencies regardless of distance.
- Multi-head: learns multiple representation subspaces.

**Other components:**

Positional Encoding (adds order info) Layer Normalization Feed-forward networks

**Advantages:**

High parallelism during training Better performance on complex inputs Replaces RNNs entirely

## Dataset Description – ChatGPT Paraphrases

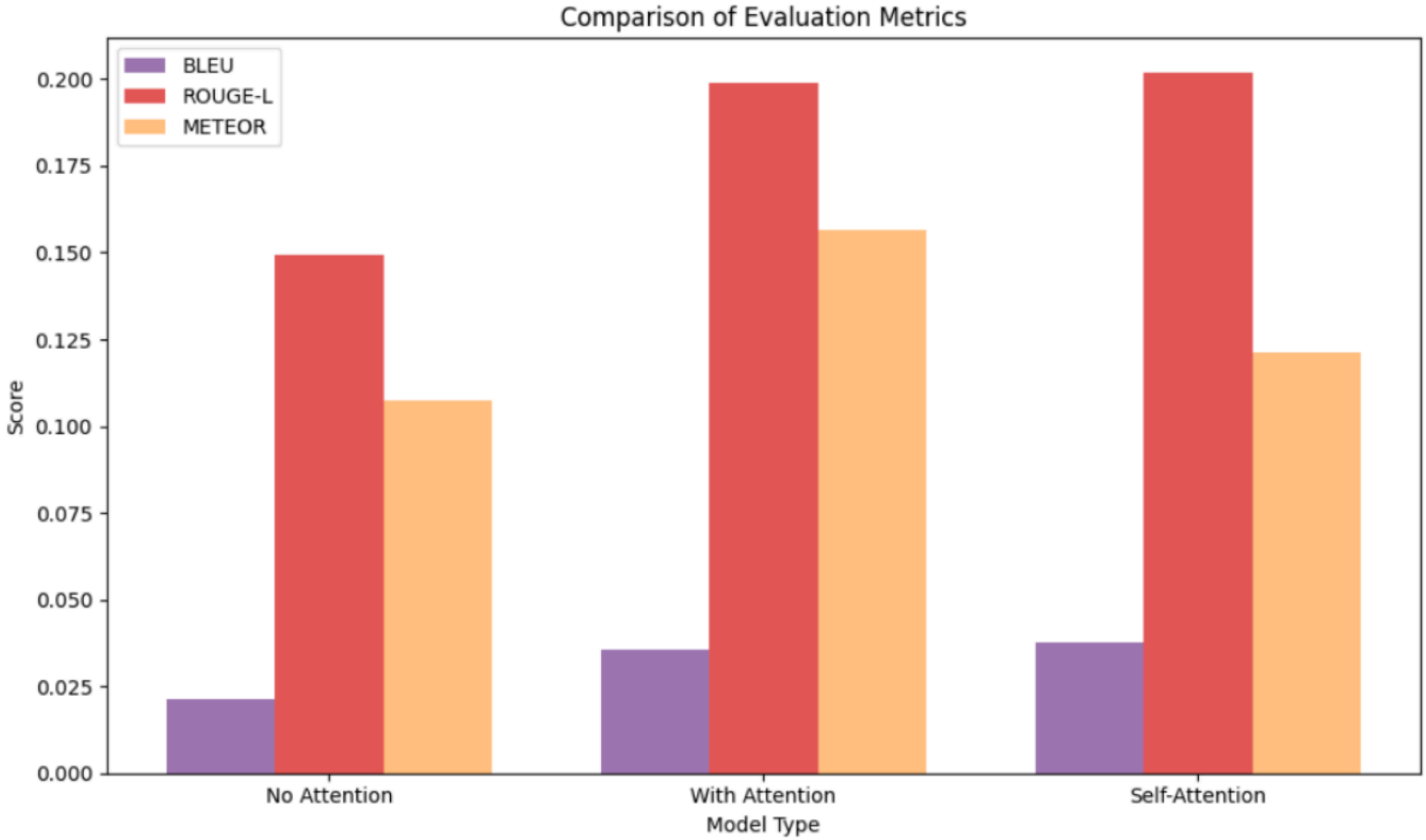
Overview:
A large-scale paraphrase dataset generated using ChatGPT, designed for training and evaluating paraphrase generation models.

Sources:
- Quora Questions – 247,138 entries
- SQuAD 2.0 Passages – 91,983 entries
- CNN News Articles – 80,076 entries

Structure:
- text: Original sentence or question
- paraphrases: List of 5 paraphrased variations
- category: Type – question or sentence
- source: Dataset origin – quora, squad_2, or cnn_news

Scale:
- ~420,000 rows total
- 5 paraphrases per row
- Up to 12.6 million paraphrase pairs can be formed
- (including bidirectional pairs)

# EVALUATION METRICS COMPARISON



# Performance Comparison

| Metrics | Without Attention | With Attention | Self Attention |
|---|---|---|---|
| Accuracy | 78.50% | 85.60% | 89.20% |
| BLEU Score | 0.41 | 0.55 | 0.63 |
| ROUGE-L | 0.5 | 0.61 | 0.69 |
| METEOR | 0.35 | 0.44 | 0.52 |
| Training Time | Fast | Moderate | High |
| Inference Speed | Fast | Moderate | Fast |
| Model Size | Small | Medium | Large |
| Interpretability | Low | High (via weights) | Moderate |

# Conclusion

- Without attention: Fast and simple, but struggles with long sequences.
- With attention: Improves quality, especially for longer/complex sentences.
- Self-attention (Transformer): Best performance; highly parallel; but more resource-intensive.
- Trade-offs: Simplicity vs Accuracy Interpretability vs Complexity

# References


- Vaswani et al., Attention Is All You Need, 2017.
- Bahdanau et al., Neural Machine Translation with Attention, 2014.
- TensorFlow, Keras documentation.
- NLTK, Rouge-Score, Hugging Face Transformers

# Thank you very much!

PRESENTED BY HELENE PAQUET