NLP Assignment 6

Vanilla Autoencoders:
Vanilla autoencoders are a type of neural network architecture used for unsupervised learning and dimensionality reduction. They consist of an encoder network that compresses the input data into a lower-dimensional representation (latent space) and a decoder network that reconstructs the original input from the compressed representation. The objective of a vanilla autoencoder is to minimize the reconstruction error between the input and the output, encouraging the model to learn a compact and informative representation of the data.

Sparse Autoencoders:
Sparse autoencoders are a variant of autoencoders that introduce sparsity constraints on the hidden layer activations during training. By penalizing the activation of hidden units and encouraging them to be close to zero, sparse autoencoders learn more robust and selective features from the input data. Sparse autoencoders are effective for feature learning, data denoising, and anomaly detection tasks.

Denoising Autoencoders:
Denoising autoencoders are designed to reconstruct clean data from corrupted or noisy input. During training, denoising autoencoders are presented with noisy input data and trained to reconstruct the original, clean input. By learning to remove noise and preserve useful information, denoising autoencoders can improve the robustness and generalization ability of the model.

Convolutional Autoencoders:
Convolutional autoencoders are a type of autoencoder architecture that incorporates convolutional layers in both the encoder and decoder networks. Convolutional autoencoders are well-suited for image data and can learn hierarchical representations of visual features. They leverage convolutional operations for local feature extraction and pooling operations for spatial downsampling, enabling effective compression and reconstruction of high-dimensional image data.

Stacked Autoencoders:
Stacked autoencoders, also known as deep autoencoders or multi-layer autoencoders, consist of multiple layers of encoder and decoder networks stacked on top of each other. Each layer learns increasingly abstract representations of the input data, leading to more powerful feature learning and representation capabilities. Stacked autoencoders are trained layer by layer using unsupervised pre-training followed by fine-tuning using backpropagation.

Generating Sentences using LSTM Autoencoders:
LSTM autoencoders can be used to generate sentences by training the model to encode and decode sequences of words. During training, the LSTM autoencoder learns to compress the input sentence into a fixed-size latent representation and then reconstruct the original sentence from the latent representation. To generate sentences, the trained decoder network is used to generate words sequentially based on the latent representation sampled from a prior distribution (e.g., Gaussian distribution).

Extractive Summarization:
Extractive summarization is a text summarization technique that involves selecting and extracting important sentences or passages from the original text to create a concise summary. Extractive summarization methods typically use algorithms to score and rank sentences based on criteria such as relevance, importance, and informativeness. The top-ranked sentences are then included in the summary without modification.

Abstractive Summarization:
Abstractive summarization is a text summarization technique that involves generating a summary by paraphrasing and synthesizing information from the original text. Unlike extractive summarization, which selects and rearranges existing sentences, abstractive summarization methods generate new sentences that capture the main ideas and concepts of the text in a more concise and coherent form. Abstractive summarization often requires natural language generation techniques, such as sequence-to-sequence models with attention mechanisms.

Beam Search:
Beam search is a search algorithm commonly used in sequence generation tasks, such as machine translation and text generation. It explores multiple candidate sequences simultaneously and maintains a fixed-size set of the most promising sequences, known as the beam. At each step, beam search expands the beam by generating possible continuations for each candidate sequence and selects the top-k sequences based on a scoring function. Beam search continues until a termination condition is met or a maximum sequence length is reached.

Length Normalization:
Length normalization is a technique used in sequence generation tasks to mitigate the bias towards shorter sequences during beam search. In length normalization, the scores of candidate sequences are divided by their lengths raised to a power (e.g., length^alpha), where alpha is a hyperparameter. This normalization penalizes shorter sequences and encourages the generation of longer, more diverse sequences.

Coverage Normalization:
Coverage normalization is a technique used in abstractive summarization to address the issue of repetition and redundancy in generated summaries. Coverage normalization tracks the attention weights assigned to each input token by the decoder during sequence generation. By penalizing high attention weights and encouraging uniform coverage of input tokens, coverage normalization helps produce more coherent and diverse summaries with reduced repetition.

ROUGE Metric Evaluation:
ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used for evaluating the quality of text summarization systems. ROUGE measures the overlap between the generated summary and reference summaries using various criteria, including n-gram overlap (ROUGE-N), longest common subsequence (ROUGE-L), and word overlap (ROUGE-W). ROUGE scores indicate the similarity and effectiveness of a generated summary compared to human-written reference summaries, with higher scores indicating better performance. ROUGE metrics are widely used in research and benchmarking for automatic summarization systems.