Text processing in Natural Language Processing (NLP) involves a series of techniques and methods used to prepare and manipulate text data for various NLP tasks like language translation, sentiment analysis, text summarization, and more.

1. **Tokenization:** This is the process of breaking down text into smaller units called tokens. Tokens can be words, characters, or subwords. This step is fundamental for most NLP tasks as it converts unstructured text into a structured form.

2. **Normalization:** This step involves converting text into a more uniform format. It includes processes like converting all characters to lower or upper case, removing punctuation, or converting numbers to words. This helps in reducing the complexity of the text data.

3. **Stop Words Removal:** Stop words are common words like "is", "and", "the", etc., that are often removed from the text as they usually don't contribute much to the meaning of a sentence for many NLP tasks.

4. **Stemming and Lemmatization:** Both are techniques to reduce words to their base or root form. Stemming crudely chops off prefixes and suffixes (e.g., "running" to "run"). Lemmatization, on the other hand, involves a more sophisticated linguistic approach to convert a word to its base or dictionary form (e.g., "better" to "good").

5. **Part-of-Speech Tagging:** This is the process of identifying parts of speech (like nouns, verbs, adjectives) in the text. It's useful in many NLP applications such as parsing and word sense disambiguation.

6. **Named Entity Recognition (NER):** NER involves identifying and categorizing key information (names of people, organizations, locations, etc.) in text. This is particularly useful in information extraction tasks.

7. **Syntactic Parsing:** This involves analyzing the grammatical structure of a sentence, identifying relationships between words, and constructing a parse tree. This is crucial for understanding the syntactic structure of sentences.

8. **Semantic Analysis:** This step involves understanding the meaning and interpretation of words in context. It includes tasks like word sense disambiguation and semantic role labeling.

9. **Feature Extraction:** This involves converting text into a set of features (numerical or categorical) for use in modeling. Techniques include Bag-of-Words, TF-IDF (Term Frequency-Inverse Document Frequency), and word embeddings like Word2Vec or GloVe.

10. **Word Embeddings:** These are vector representations of words that capture contextual meanings, syntactic and semantic relationships. Models like Word2Vec, GloVe, and BERT provide pre-trained embeddings which can be used for various NLP tasks.

11. **Text Classification and Clustering:** Classification involves categorizing text into predefined classes, while clustering involves grouping similar texts together. These are common tasks in NLP for applications like spam detection, sentiment analysis, and topic modeling.

Deep Learning:

Definition: Deep learning is a subset of machine learning that uses artificial neural networks with multiple layers (deep architectures) to model and understand complex patterns in large datasets.

Neural Networks: Deep learning primarily relies on neural networks, which are computational models inspired by the structure and function of biological neurons in the human brain.

Layers: Deep learning models consist of multiple layers, including input layers, hidden layers, and output layers. These layers perform various transformations on the input data to learn representations at different levels of abstraction.

Activation Functions: Activation functions introduce non-linearities into the network, enabling it to model complex relationships in the data.Common activation functions include sigmoid, tanh, ReLU, and softmax.

Training Algorithms: Deep learning models are trained using algorithms such as backpropagation, stochastic gradient descent, and their variants to optimize the network parameters (weights and biases) based on the difference between predicted and actual outputs.

Weights and Biases: Neural networks are parameterized by weights and biases, which are learned during the training process. Weights determine the strength of connections between neurons, while biases allow the network to learn non-zero intercepts.

Optimizers: Algorithms used to update the weights and biases of the neural network during training to minimize the loss function. Examples include stochastic gradient descent (SGD), Adam, RMSprop, and Adagrad.

Loss Function: Measures the difference between the predicted output of the network and the actual output (the ground truth). Common loss functions include mean squared error (MSE), cross-entropy loss, and binary cross-entropy loss.

Forward Propagation: Involves passing input data through the neural network to compute the predicted output.
Each layer performs a transformation on the input data based on its weights, biases, and activation function.
The output of one layer becomes the input to the next layer until the final output is produced.

Backward Propagation: Involves computing the gradients of the loss function with respect to the weights and biases of the network.Gradients are computed using the chain rule of calculus and are used to update the network parameters during training.Backpropagation allows the network to learn from its mistakes by adjusting the weights and biases to minimize the loss function.

Feedforward Neural Networks (FNNs): Basic type of neural network where information flows in one direction, from input nodes through hidden layers to output nodes.
Typically used for regression and classification tasks.

Convolutional Neural Networks (CNNs): Specialized for processing grid-like data such as images.Consist of convolutional layers for feature extraction and pooling layers for down-sampling.Widely used in image recognition, object detection, and image segmentation tasks.

Recurrent Neural Networks (RNNs): Designed to handle sequential data with temporal dependencies.Employ recurrent connections that allow information to persist over time.Suitable for tasks like natural language processing (NLP), time series prediction, and speech recognition.

Long Short-Term Memory Networks (LSTMs) and Gated Recurrent Units (GRUs): Specialized types of RNNs designed to address the vanishing gradient problem and capture long-range dependencies in sequential data.

Widely used in tasks requiring memory over long sequences, such as machine translation and speech synthesis.

Autoencoders: Neural networks trained to reconstruct input data at the output layer.Consist of an encoder network that compresses the input into a latent representation and a decoder network that reconstructs the input from the latent representation.Used for data compression, feature learning, and anomaly detection.

Generative Adversarial Networks (GANs): Consist of two neural networks, a generator and a discriminator, trained simultaneously in a game-theoretic framework.The generator generates synthetic data samples, while the discriminator distinguishes between real and fake samples. Widely used for generating realistic images, data augmentation, and unsupervised learning.