Research Paper Preliminary Information

1. Large language models in machine translation

Backoff smoothing: Backoff smoothing is a technique used in NLP to smooth the probabilities estimated by language models. The idea behind backoff smoothing is to estimate the probability of an n-gram based on the probabilities of lower-order n-grams if the higher-order n-gram has not been seen in the training data. For example, if the probability of a bigram (e.g., "cat dog") cannot be estimated from the training data, the probability of the unigram "cat" may be used instead. If the unigram "cat" has not been seen in the training data either, the language model can fall back to a default probability based on some other smoothing method. Backoff smoothing can help to address the problem of zero probabilities in language models and prevent overfitting to the training data.

Information loss in two-pass decoding: In two-pass decoding, the first pass generates an initial hypothesis, and the second pass refines the hypothesis based on additional information. Information loss in two-pass decoding occurs because some information from the first pass is discarded during the second pass, which can lead to a reduction in the quality of the final output. For example, the first pass may generate a hypothesis that is grammatically correct but semantically incorrect, and the second pass may not be able to correct this error because the information that was available in the first pass is lost.

Kneser-Ney smoothing: Kneser-Ney smoothing works by estimating the probability of an n-gram based on the number of times it has been seen in the training data and the number of times its lower-order n-grams have been seen. This allows the model to more accurately capture the context in which the n-gram occurs and to avoid overfitting to the training data.

BLEU score: The BLEU (Bilingual Evaluation Understudy) score is a common evaluation metric for machine translation. It measures the overlapping n-grams (sequence of words) between the predicted translation and the reference translation and assigns a score based on the number of matches. Higher BLEU scores indicate a higher degree of similarity between the predicted and reference translations, and BLEU scores closer to 1.0 are considered better. It is a commonly used metric, but has limitations and has been criticized for not always accurately reflecting the quality of a machine translation.

2. Attention is all you need

Recurrent Neural Networks: Recurrent Neural Networks (RNNs) are a type of deep learning model that are used for processing sequential data such as speech, text, and time series. Unlike traditional neural networks, RNNs have a "memory" mechanism in the form of hidden states that allow them to retain information from previous time steps, making them well-suited for tasks such as language modeling, machine translation, and speech recognition. The hidden states are computed based on the current input and the previous hidden state, and this process is repeated for every time step in the sequence. There are various types of RNNs, including Vanilla RNNs, Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs), which vary in terms of the type of hidden state used and the mechanism for controlling information flow through the hidden states.

Self-attention mechanisms: Self-attention mechanisms are a type of attention mechanism used in deep learning models to dynamically weight the importance of different elements in a sequence of data. They allow the model to focus on relevant parts of the input and produce more context-aware representations. In self-attention, each element in the sequence is first transformed into a query, key, and value representation. The model then computes attention scores between each query and key and uses these scores to weigh the values to compute a weighted sum, which becomes the new representation of the input sequence. This process is repeated multiple times to produce the final representation. Self-attention has been successfully used in various tasks such as machine translation, text classification, and image captioning. It has become a popular building block for Transformer-based models, which have achieved state-of-the-art results in a variety of NLP tasks.

Multi-headed self-attention mechanisms: It allows the model to attend to multiple parts of the input sequence in parallel, thus capturing more fine-grained information from the input. In multi-head self-attention, the input sequence is transformed into multiple sets of queries, keys, and values, each representing a different aspect of the input. Attention scores are computed between each query and key, and the values are then weighed and summed to produce multiple attention outputs. These outputs are concatenated and passed through a feedforward layer to produce the final representation.

Residual connection: The idea behind residual connections is to make it easier for the network to learn complex functions by allowing the gradients to flow more easily through the network during backpropagation. The residual connection enables the network to learn the residual or the difference between the desired output and the current output, rather than learning the entire function from scratch.

Key-value pairs in self-attention: The key-value pairs in a self-attention mechanism provide a way to weigh the importance of each element in the input sequence, which allows the model to attend to relevant parts of the input and produce context-aware representations. Key is the similarity score between input elements, and value is the information about each element.

3. Improving language understanding by generative pre-training

Token embedding matrix: A token embedding matrix is a matrix in a natural language processing (NLP) model that represents words or subwords (tokens) as numerical vectors (embeddings). Each row in the matrix corresponds to a unique token, and each column represents a dimension of the token's representation. The values in the token embedding matrix are learned during training, typically using an optimization algorithm such as stochastic gradient descent. The goal is to learn meaningful representations that capture the relationships between words in the language.

Zero-shot learning: Zero-shot learning is a machine learning setting in which the model is expected to classify or predict instances of classes that it has never seen during training. In other words, the model must generalize its knowledge to new, unseen classes based on its prior experience with other classes.

4. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Cloze task: The Cloze task is a type of natural language processing (NLP) task that involves predicting missing words in a sentence or a text. In the Cloze task, a portion of the sentence or text is removed and replaced with a placeholder, such as [MASK]. The goal is to predict what word or phrase should be inserted in place of the [MASK]. The Cloze task is typically used as a way to evaluate the language understanding capabilities of NLP models. For example, a model trained on a large corpus of text can be evaluated on its ability to predict the missing words in a Cloze task. The performance of the model can be evaluated based on its accuracy in predicting the correct words. Also known as MLM (masked language model).

Downstream task: Downstream tasks refer to specific NLP tasks that are applied to a pre-trained model to fine-tune it for a specific task. These tasks are called "downstream" because they are applied after the model has been pre-trained on a large corpus of data. For example, a pre-trained language model such as BERT can be fine-tuned for downstream tasks such as sentiment analysis, question answering, or text classification. In this process, the model is fine-tuned on a smaller, task-specific dataset to adapt it to the specific task at hand. This fine-tuning process can improve the performance of the model on the downstream task compared to using the pre-trained model directly.

5. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context

Vanilla transformers: A vanilla Transformer is a standard Transformer model with no additional bells and whistles. It has the standard structure, consisting of an encoder and decoder, multi-head attention mechanism, feed-forward network, and residual connections, as described in the original paper "Attention is All You Need" by Vaswani et al. However, researchers and practitioners often modify the Transformer architecture to improve performance, and these modified versions are not considered "vanilla." Some examples include the BERT (Bidirectional Encoder Representations from Transformers) model.

Perplexity: Perplexity is a common evaluation metric used in Natural Language Processing (NLP) to measure the quality of a language model. It provides an estimate of how well a language model predicts a sample of text by comparing its predictions to the actual text. In mathematical terms, perplexity is defined as the exponentiation of the cross-entropy loss between the predicted probabilities and the true probabilities of the words in the text. The cross-entropy loss is used to measure the difference between the predicted probabilities and the actual probabilities, and the exponentiation makes the result a more intuitive measure of the model's quality.

6. Language Models are Unsupervised Multitask Learners

ROUGE: It calculates the overlap between the generated summary and the reference summary in terms of n-grams, where n can be 1 (unigrams), 2 (bigrams), or 3 (trigrams). The most common ROUGE metric is ROUGE-N, which calculates the recall of the n-grams in the generated summary with respect to the reference summary. There are several variants of ROUGE, including ROUGE-1, ROUGE-2, and ROUGE-L, which differ in the way they calculate the overlap between the generated and reference summaries. For example, ROUGE-L measures the longest contiguous sequence that appears in both the generated and reference summaries.

7. Language Models are Few-Shot Learners

Few-shot learning: Few-shot learning is a type of machine learning that focuses on learning from a small number of examples. It is a crucial problem in artificial intelligence and machine learning, as real-world applications often involve learning from very few examples, or even from just one example. In few-shot learning, the goal is to learn a model that can generalize to new classes based on only a few examples of each class.

Beam search: Beam search is a search algorithm that is commonly used in Natural Language Processing (NLP) and other domains. It is a heuristic search algorithm that explores a search space by maintaining a set of k-best candidates at each step, where k is a user-defined parameter known as the beam width. In a beam search, the algorithm starts with an initial state, and at each step, it generates a set of k-best candidates by applying a set of possible actions. The set of k-best candidates is then used to generate the next set of k-best candidates, and this process is repeated until a stopping criterion is met.

Length penalty in beam search: The idea behind length penalty is to reward shorter sequences with higher likelihood, while penalizing longer sequences with lower likelihood. This helps to balance the trade-off between generating short, high-quality sequences and generating longer, lower-quality sequences.

8. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

Distillation: It is a powerful technique for compressing large pre-trained transformer models into smaller, more efficient models, while maintaining their accuracy and generalization ability. It allows for the deployment of these models on resource-constrained devices and can make them more adaptable to new tasks and domains.

Soft targets: Soft targets in multi-class classification refer to the use of class probabilities as targets instead of hard labels (i.e., one-hot encoded vectors). Instead of using a one-hot encoded vector to represent the true label for each instance, soft targets use a probability distribution over the classes, where the probability of the true label is higher than the probabilities of the other classes.