Modelling = Unit-4

Language Modelling - Introduction, n-Gram Models, Language model Evaluation, Parameter Estimation, Language Model Adaption, Types of Language Models, Language-Specific Modelling Problems, Multilingual and Cross lingual Language Modelling.

Language Modeling :-

Language modeling is a foundational task in NLP that involves predicting the next word in sequence of words in a text given the words that precede it.

This predictive capability enables various applications such as autocomplete suggestions, machine translation, speech recognition, sentiment analysis, and text generation.

Probability and Prediction:

Language models estimate the probability of a word given its content.

Types of Language Models:

Statistical Models: Traditional models like n-grams and HMMs - Hidden Markov Models use statistical methods to predict the next word based on frequencies and probabilities derived from training data.

Neural Nlw Models: Modern language models, particularly those based on DL, employ neural networks such as RNN, LSTM, and Transformers.

These models can capture complex dependencies and semantic relationships in language.

Applications:

Text Generation: Language models can generate

human-like text, which is useful in appⁿ like chatbots, content ~~generation~~ creation, and creative writing assistance.

Speech Recognition: Models predict spoken words based on acoustic signals.

Machine Translation: Models translate text from one language to another by predicting the most likely translations.

Training and Evaluation:

Training: Models are trained on large datasets of text, adjusting their parameters to minimize prediction errors and maximize accuracy.

Evaluation: Models are evaluated based on metrics like perplexity and BLEU score.

State of the Art Models:

Recent advancements include models like GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers)

n-Gram Models:-

n-gram models are a type of probabilistic language model used in NLP and computational linguistics.

They predict the likelihood of a word based on the previous n-1 words in the sequence.

Defⁿ: An n-gram is a contiguous sequence of n-items (words, characters, or tokens) from a given text.

eg: I love NLP.

2-grams (bigrams) ⇒ I love, love NLP.

Markov Assumption:

n-gram models operate under the Markov assumption,

which states that the probability of a word only depends on the previous n-1 words.

Probability Estimation:

n-gram models estimate the probability of a word sequence using relative frequencies observed in training corpus.

$$P(w_i | w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$$

Smoothing:

Since n-gram models can encounter unseen n-gram (i.e, combinations of words not present in the training data), smoothing techniques like Laplace smoothing or Good-Turing smoothing are often applied to handle such cases and prevent zero probabilities.

Applications: n-gram models are used in various NLP tasks such as:

- Language Modeling
- Speech Recognition.
- Machine Translation.

Limitations:

n-gram models have limited context sensitivity due to their fixed window size 'n'. They struggle with capturing long-range dependencies and understanding semantic relationships b/w distant words.

Language Model Evaluation:-

Language model evaluation is essential to assess the performance and effectiveness of models in NLP tasks.

Metrics for Evaluation:

Perplexity: It is a common metric used to

evaluate language models, especially statistical and NN-based models.
Lower perplexity indicates better performance.

BLEU: Bilingual Evaluation Understudy
BLEU is commonly used for evaluating the quality of machine-translated text by comparing it to one (or more) reference translations.

ROUGE: Recall-Oriented Understudy for Gisting Evaluation.
ROUGE evaluates the quality of text summaries by comparing them to reference summaries. It measures precision, recall, and F1 score for overlapping n-grams b/w the generated and reference summaries.

WER: Word Error Rate
WER is used in speech recognition tasks to evaluate the accuracy of transcriptions.

Accuracy and Precision: These metrics are used in specific tasks like sentiment analysis, NER, and POS tagging to evaluate the correctness of predictions.

Parameter Estimation:-
Parameter estimation in the context of language modeling refers to the process of determining the optimal values for the parameters of a probabilistic model based on observed data.

MLE: Maximum - Likelihood Estimation.
It is a common method used to estimate the parameters of a statistical model.

MDI: Minimum Discrimination Information.
MDI is an alternative criterion for parameter estimation in language models, especially for NN-based models.

Regularization:
To prevent overfitting and improve the generalization ability of language models.

Cross-Validation:
To evaluate the performance of parameter estimation methods and select optimal hyperparameters.

Bayesian Parameter Estimation:-
Bayesian parameter estimation is an approach to parameter estimation that incorporates prior knowledge (or) beliefs about the parameters in to the estimation process.

Bayes Theorem:
It forms the basis of Bayesian inference and parameter estimation.

$$P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)}$$

The prior distribution $P(\theta)$ encapsulates prior beliefs to knowledge about the parameters θ before observing the data.
The posterior distribution $P(\theta|D)$ represents the updated beliefs about the parameters θ after observing the data D.

Advantages:
→ Incorporation of Prior knowledge.
→ Uncertainty Quantification.
→ Flexibility.

Challenges:
→ Choice of priors
→ Computational Complexity.
→ Interpretability.

Language Model Adaption:-

Language model Adaption refers to the process of fine-tuning a pre-existing language model to better perform a specific task (or to better understand a particular domain or content.

Task-Specific fine-tuning:

Adaptation typically involves re-training a pre-trained language model on task-specific data to improve its performance on a specific task.

eg: sentiment analysis, NER, MT, document classification

Domain Adaptation:

Adapting a language model to understand and generate text specific to a particular domain.

eg: medical, legal, financial.

Data Augmentation:

Increasing the diversity and volume of training data by incorporating additional relevant data.

It helps to improve the robustness and generalization of the adapted language model.

Transfer Learning:

Leveraging knowledge learned from a pre-trained model to adapt to new tasks (or domains with limited annotated data.

Fine Tuning Procedure:

Typically involves initializing the pre-trained model with weights from a general model (eg. BERT and GPT), and fine tuning it using task-specific or domain specific labeled data through gradient-based optimization techniques like SGD or Adam.

Evaluation and Iteration:

After fine-tuning, the adapted language model is evaluated on a validation set using cross-validation to assess its performance.

Iterative refinement may be necessary to achieve desired performance metrics, adjusting hyper-parameters or incorporating additional data as needed.

Applications:

→ Customized chatbots.
→ Medical NLP
→ Legal and Compliance.

Ethical Considerations:

→ Bias and fairness
→ Privacy.

Language model adaption plays a crucial role in extending the utility of pre-trained models to specific tasks and domains, enhancing their practical applicability and performance in real-world applications.

Types of Language models:-

There are several types of language models used in NLP, each with its own characteristics and applications.

Sequence-to-Sequence Models:

These models use an encoder-decoder architecture to map I/p sequences to O/p sequences, often used in tasks like MT, summarization, and dialogue generation.

Transformer-based Models:

Transformers use self-attention mechanisms to capture global dependencies and relationships across i/p sequences, making them highly effective for modeling long-range dependencies.

PCFG: probabilistic CFGs.

PCFGs extend CFGs by assigning probabilities to production rules, capturing statistical properties of language syntax.

They are used in syntactic parsing and grammar checking applications.

Ensemble Models:

Ensemble models combine predictions from multiple base models to improve overall performance and robustness.

Deep Generative Models:

These models generate new data samples by learning the underlying distribution of the training data, often using techniques like VAEs - Variational Auto Encoders and GANs - Generative Adversarial Nws.

They are used in tasks like text generation, style transfer, and data augmentation.

Class Based Language Models:-

Class-based language models are a specialized approach to language modeling that categorizes words (or tokens) into classes based on their usage patterns (or semantic similarities).

Instead of predicting individual words directly, these models predict classes of words, which can improve efficiency and generalization.

Word clustering:

Class-based models cluster words into groups based on similarities in their context or usage patterns.

Class-based N-gram Models:

In this, n-grams are represented by sequences of class labels instead of individual words.

Hierarchical Softmax:

It is a method used in class-based models to efficiently handle large vocabularies by organizing words into a hierarchical structure.

Class-based Neural Language Models:

NN based language models can be adapted to use class embeddings instead of word embeddings learning representations that capture semantic or syntactic similarities among classes.

Advantages:

→ Generalization.
→ Efficiency.

Applications:

→ Low-Resource Languages
→ Speech Recognition.

Challenges:

→ Class Definition.
→ Model Training.

Variable Length Language Models:-

Variable length language models refer to models that can handle input and output sequences of varying lengths, rather than fixed-length sequences.

Sequence-to-Sequence Models:

Seq2Seq models use an encoder-decoder architecture to map variable-length i/p sequences to variable-length o/p sequences.

They are widely used in MT, text summarization and dialogue generation.

Attention Mechanisms:

Attention mechanisms are used in Transformer models, enable the model to focus on relevant parts of the i/p sequence when generating each part of the o/p sequence.

Beam Search and Decoding Strategies:

Beam search is a decoding strategy used to generate o/p sequences by exploring multiple hypotheses and selecting the most probable sequence.

Dynamic Input and Output Handling:

Variable length models dynamically adjust to the length of i/p sequences during training and inference.

Challenges:

→ Training Complexity.
→ Decoding efficiency.

Advantages and Apps:

→ Document Understanding
→ Conversation Generation.

Discriminative Language Models:-

Discriminative language models are a type of language model that focuses on predicting a specific outcome or label directly from i/p data, rather than modeling the entire probability distribution of the i/p sequence itself.

Task-Specific Prediction:

Discriminative models are designed to predict specific outcomes or labels directly relevant to a task, such as sentiment classification, NER, MT.

Conditional Models:

These models estimate $P(y|x)$, where 'x' represents the i/p and 'y' represents output.

Feature Engineering:
Discriminative models often require carefully engineered features that are relevant to the task at hand.
These features can include linguistic features, syntactic structures, or domain specific characteristics.

Examples:
→ Logistic Regression
→ SVM.

Advantages:
→ Efficiency
→ Scalability.

Limitations:
→ Limited Information Capture
→ Dependency on features.

Applications:
→ Text classification
→ NER. → MT

Syntax-Based Language Models:-
These models that utilize syntactic structures and grammar rules to understand and generate NL text.
These models incorporate linguistic principles to improve accuracy and coherence in tasks such as parsing, text generation, and NER. MT.

Syntactic Parsing:
Syntax-based models analyze the grammatical structure of sentences by parsing them into syntactic constituents, based on grammar rules.

Grammar Rules:
These models incorporate predefined grammar rules or syntactic constraints to guide the generation or understanding of text.

Contextual Understanding:
Syntax-based models leverage syntactic information to capture deeper semantic relationships and dependencies b/w words and phrases in a sentence.

Syntax-Aware Neural Models:

NN architectures, such as RNN, Tree-LSTM (or graph NN, can integrate syntactic information to improve learning & prediction tasks.

Applications:

→ Text Generation
→ Translation Quality.
→ Parsing and Information Extraction.

Challenges:

→ Ambiguity.
→ Training data.

Maxtnt Language Models:-

Maximum Entropy (Maxtnt) language models, also known as MEMMs - Maxtnt Markov Models, are probabilistic models used in NLP to predict sequences of events, particularly in sequence labeling tasks such as POS tagging and NER.

Principle of Maxtnt:

Maxtnt models aim to select the probability distribution that maximizes entropy (uncertainty) while satisfying a set of constraints derived from the observed data.

Markov Property:

MEMMs extend Maxtnt models by incorporating the Markov property, which assumes that the probability of each event depends only on the preceding event.

Feature Based Representation:

Maxtnt models use feature functions that capture relevant characteristics or patterns in the i/p data.

Training:

Maxtnt models are typically trained using

MLE & maximum a posteriori (MAP) estimation, optimizing the model parameters to best fit the observed data.

Applications:
→ Sequence Labeling
→ NER

Advantages:
→ Flexibility
→ Interpretability.

Challenges:
→ Feature Engineering
→ Scalability.

Factored Language Models:-

These models that enhances traditional n-gram models by incorporating additional factors & attributes beyond just the previous n-words. These factors can include syntactic information, semantic features, & other contextual clues that improve the models ability to predict the next word & sequence in NLP tasks.

Factorization of Features:

Factored language models decompose the prediction task into multiple factors & dimensions, each representing diffent aspects of the content in i/p sequence.

Factor Graphs:

Factored language models are often represented using factor graphs, which model dependencies among variables (factors) and facilitate efficient inference and learning.

Integration of Linguistic Knowledge:

By incorporating syntactic and semantic factors, factored language models improve the accuracy

and robustness of predictions compared to traditional n-gram models.

Inference: During inference, the model combines predictions from different factors to compute the most likely sequence of words or labels, often using probabilistic inference methods.

Advantages:
→ Contextual sensitivity.
→ Generalization.

Challenges:
→ Complexity
→ Feature Engineering.

Applications:
→ NLU → Speech Recognition.

Other Tree Based Language Models:-

Tree-based language models refer to models that utilize tree structure to represent and process NL.

These models leverage hierarchical representations of language, capturing syntactic dependencies and semantic relationships more explicitly compared to linear sequence-based models like n-grams or NN.

RecNN:

RecNN, recursively apply the same NN function to parse tree structures of sentences, capturing hierarchical dependencies.

Tree-Structured LSTMs / NNs:

Tree-LSTMs extend standard LSTMs by incorporating tree structures, allowing for more direct modeling of hierarchical relationships in language.

Graph-Based NN:

These models represent sentences as graphs, where nodes correspond to words or tokens connected by edges indicating syntactic/semantic

relationships.

Dependency - Based Models:

Dependency parsers model syntactic relationships as directed dependencies b/w words, representing sentences as dependency trees.

CCG: Combinatory Categorial Grammar.

CCG is a formalism that combines syntactic and semantic composition rules to generate parse trees, emphasizing the combination of lexical categories and syntactic structures.

Applications:

→ Semantic Parsing
→ Dialogue systems
→ Interpretability.

Bayesian Topic-Based Language Models:-

These are probabilistic models that combine Bayesian inference with topic modeling techniques to discover latent topics from a corpus of text data.

These models are particularly useful for uncovering thematic structures in large text collections and facilitating tasks such as document clustering, information retrieval, and content recommendation.

Probabilistic Topic Models:

Bayesian topic models, such as LDA - Latent Dirichlet Allocation and its extensions, assume that each document is a mixture of latent topics, and each topic is characterized by a distribution over words.

Generative Process:

In this, each document is generated by sampling topics according to a document-specific distribution, and words are generated based on the topic-word distributions.

Dirichlet Priors:

Dirichlet priors are used in Bayesian topic models to model the distribution of topics and words.

Variational Inference:

Variational methods approximate the posterior distribution over latent variables, such as topics, by optimizing a lower bound on the model's marginal likelihood.

Gibbs Sampling:

Gibbs sampling iteratively samples values for latent variables from their conditional distributions, converging to the posterior distribution over topics.

Applications:

→ Document clustering
→ Information Retrieval
→ Content Recommendation.

NN Language Models:-

NNLM are a class of language models that use NN to learn the probability distribution of sequences of words in NL.

These models have gained popularity due to their ability to capture complex patterns and dependencies in text data, surpassing traditional n-gram models in various NLP tasks.

Feed forward NN: Basic NNLMs consist of a feed forward NN that learns to predict the probability of the next word given a fixed-size window of previous words.

RNN: RNN-based language models use recurrent connections to capture sequential dependencies over variable-length contexts, allowing them to model longer-term dependencies in text.

Transformers: Transformer-based models, such as GPT and BERT leverage self-attention mechanisms to capture global dependencies and improve performance on diverse NLP tasks.

Word Embedding: NNLMs typically use word embeddings to convert words into dense, continuous vector representations. These embeddings capture semantic relationships and contextual meanings of words.

Backpropagation: Training involves backpropagation and gradient descent algms to adjust model parameters based on prediction errors, optimizing the model for better language modeling performance.

Applications:
- Text Generation
- Language Understanding.

Advantages:
- Capturing Contextual Information.
- Flexibility.

Challenges:
- Data efficiency
- Computational Resources.

Language Specific Modeling Problems:-

Language specific modeling problems in NLP arise due to unique characteristics, structures, and challenges inherent to different languages. These problems often require tailored approaches and solutions to effectively process and understand text in specific linguistic contexts.

Morphological Complexity:

Languages with rich morphology pose challenges in tokenization, stemming, and word normalization due to complex word forms and grammatical variations.

Word Order and Syntax:

Languages vary significantly in word order and syntactic structures, influencing tasks like parsing, dependency analysis, and syntactic parsing accuracy.

Resource Scarcity:

Low-resource languages lack sufficient annotated data, pre-trained models, and linguistic resources, hindering the development and performance of NLP systems.

Orthographic Variation:

Languages with diverse writing systems, characters, and orthographic conventions pose challenges in text normalization, tokenization, and handling of typographical variations.

NER:

NER systems need to handle diverse named entity types and their linguistic variations across languages, such as person names, locations, and organizations.

Sentiment Analysis and Cultural Nuances:
Sentiment analysis models must account for cultural differences, expressions and contextual nuances in sentiment polarity and opinion mining across languages.

Code-Switching and Multilingualism:
Languages with frequent code-switching and multilingual text present challenges in language identification, translation, and sentiment analysis.

Dialectal Variation:
Variations in dialects and regional language variations impact language modeling, speech recognition, and text-to-speech, requiring dialect-specific models.

Language Modeling for Morphologically Rich Languages:
Language modeling for morphologically rich languages presents unique challenges and opportunities due to their complex word forms, extensive inflectional patterns, and rich morphological structures.

Word Tokenization and Segmentation:
Morphologically rich languages often have complex word forms with multiple affixes and morphemes, making tokenization and segmentation challenging.
Use morphological analyzers and tokenizers that can decompose words into morphemes or subword units to handle variations in word forms.

Data Sparsity:
Augment training data with techniques like data synthesis, cross-lingual transfer learning &

leveraging unsupervised pre-training methods to improve model robustness.

Morphological Ambiguity:

Words in morphologically rich languages often exhibit ambiguity due to multiple possible morphological analyses or interpretations.

Incorporate contextual information and linguistic features into language models to disambiguate word meanings and improve prediction accuracy.

OOV Words: Out of vocabulary.

Morphologically rich languages have a large vocabulary size and higher rates of OOV words compared to less morphologically complex languages

Employ subword or character level modeling techniques to handle unseen words and improve generalization.

Modeling Long-Term Dependencies:

Capturing long-range dependencies and context in morphologically rich languages requires models that can effectively handle variable-length contexts and complex syntactic structures.

Selection of subword units:-

Selection of subword units is crucial in NLP tasks, particularly for handling morphologically rich languages, improving OOV word handling, and enhancing model generalization.

Byte-Pair Encoding (BPE):

BPE algm iteratively merges the most frequent pair of consecutive characters or bytes to create a vocabulary of subword units.

Word Piece:

Word Piece algm, similar to BPE, iteratively merges the most probable pair of subword units based on a language model's likelihood.

ULM: Unigram Language Model.

ULM tokenization builds a vocabulary of subword units by training a language model to predict the next subword given the content.

Character-Level Models:

Model tokens as sequences of characters rather than predefined subword units, allowing the model to handle any word form.

Benefits:

→ Improved Generalization
→ Enhanced Efficiency
→ Linguistic Insight

Applications:

→ POS Tagging
→ Morphological Generation
→ Semantic Parsing.

Challenges:

→ Annotation Complexity
→ Ambiguity
→ Integration with models.

Spoken Vs Written Languages:-

Spoken and written languages exhibit distinct characteristics in terms of structure, grammar, vocabulary usage and communication norms.

Spoken Languages:

Informality and Fluidity:

Characteristics: Spoken languages tends to be more informal, with greater flexibility in grammar, sentence structure, and word choice.

Prosody and Intonation:
Communication relies heavily on intonation, stress, pitch variations, and rhythm to convey meaning, emotions, and emphasis.
Immediate Feedback:
Interaction often involves real-time feedback, clarification, and adaptation, based on the listeners responses & non-verbal cues.
Spontaneity and Repetition:
Speakers may repeat words & phrases for emphasis, clarification & to reinforce key points during conversation.
Context Dependency:
Understanding relies on shared knowledge, situational context and programs cues that may not be explicitly stated.
Written Languages:
Formality and Structure:
Written languages tends to be more formal, adhering to grammatical rules, punctuation, and standard conventions.
Lack of Prosody:
Written communication lacks the prosodic features of spoken language, such as intonation and rhythm, relying solely on textual cues.
Permanent and Editable:
Written texts are permanent and can be revised, edited and reviewed before distribution.
Complexity and Density:
Written languages can be more complex, with

longer sentences, richer vocabulary, and specialized terminology suited for formal communication.

Autonomy from Context:

Understanding relies on textual cues and explicit information provided within the written document, rather than external context.

Multilingual Language Modeling:-

Multilingual language modeling involves developing models that can understand, generate, or process multiple languages within a single framework.

Challenges:

Language Diversity:

Languages vary in syntax, morphology, grammar and vocabulary requiring models to handle linguistic diversity effectively.

Data Availability:

Availability of labeled data varies across languages, with some languages having sparse resources or lower-quality datasets.

Code-Switching and Multilingual Contexts:

Users often mix languages within a single conversation or text, challenging models to maintain context and meaning across languages.

Evaluation and Performance:

Evaluating model performance across multiple languages requires language-specific metrics and benchmarks that reflect diverse linguistic characteristics.

Cross-lingual Transfer Learning:

Pre-train models on large-scale datasets from

one or more languages and transfer knowledge to tasks or domains in other languages.

Multilingual Embeddings:
Learn embeddings that capture semantic similarities and relationships across multiple languages..

Language Agnostic Architecture:
Design models with architectures that are adaptable to various languages without language-specific modifications.

Language Specific Fine Tuning:
Fine-tune pre-trained multilingual models on language-specific tasks or datasets to optimize performance for individual languages.

Applications:

→ MT
→ Cross-lingual Information Retrieval
→ Multilingual chatbots and Dialogue systems
→ Multilingual sentiment Analysis and Opinion Mining.
→ Language Understanding and Generation.

Crosslingual Language Modeling:-
Crosslingual language modeling focuses on developing models that can effectively understand and generate text across multiple languages, leveraging shared representations and transfer learning techniques.

Multilingual Embeddings:
Learn embeddings that map words or sentences from multiple languages into a shared semantic space.

Cross-lingual Transfer learning:
Pre-train models on data from one or more languages and transfer knowledge to tasks or domains in other languages.

Zero-shot and Few-shot learning:
Enable models to generalize to unseen languages by training on a diverse set of languages during pre-training.

Language Adversarial Training:
Train models to be invariant to language-specific variations or features by incorporating adversarial training techniques.

→

Challenges:
- → Data availability and Quality
- → Language Divergence
- → Evaluation Metrics.
- → Crosslingual Understanding and generation.

Applications:
- → Multilingual MT
- → Cross lingual IR.
- → → → → → → →
- → Crosslingual Question Answering and Summarization