LanguageModelling-Introductionandn-GramModels

The Models that assign probabilities to sequences of words are called **language models (LMs).** The simplest language model that assigns probabilities to sentences and sequences of words is the **n-gram**model.

An**n-gram**isasequenceofNwords: Forexample**—Pleaseturnyourhomework‖**canbe written as:
–A1-gram(unigram)isasinglewordsequenceofwordslike—please‖,—turn‖,—your‖, —homework‖

–A 2-gram (bigram) is a two-word sequence of words like —please turn‖, —turn your‖, or ‖your homework‖.

–A 3-gram (trigram) is a three-word sequence of words like —please turn your‖, or —turnyour homework‖.

Wecanusen-grammodelstoestimatetheprobability ofthelastwordofann-gramgiventhe previous words, and also to assign probabilities to entire word sequences.

ProbabilisticLanguageModels:

Probabilisticlanguagemodelscanbeusedtoassignaprobabilitytoasentenceinmany NLP tasks.

•Machine Translation: P(highwindstonight)>P(largewinds tonight)
•SpellCorrection: Thekofficeisabout tenminutesfrom here
P(TheOfficeis)>P(Thenofficeis)
•SpeechRecognition:
P(Isaw avan)>>P(eyesaweofan)
•Summarization,question-answering.

Ourgoalistocomputetheprobabilityofasentenceorsequenceofwords W(=w1,w2,…wn): P(W) = P(w1,w2,w3,w4,w5…wn)
•What isthe probabilityofan upcoming word?: P(w5|w1,w2,w3,w4)
•Amodelthatcomputeseitherofthese:P(W)orP(wn|w1,w2…wn-1)iscalled a**language model**.

ChainRuleofProbability

Wecomputeprobabilitiesofentirewordsequenceslikew1,w2,…wn The probability of the word sequence w1,w2,…wn is P(w1,w2,…wn).
Wecanusethe**chain ruleoftheprobability**todecomposethis probability:

$$P(w_1^n) = P(w_1)\,P(w_2|w_1)\,P(w_3|w_1^2)\,\ldots\,P(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

Example:P(themanfromjupiter) =
P(the)P(man|the)P(from|theman)P(jupiter|themanfrom)

N-Grams

The aim of the n-gram model (simplifying assumption) is to predict the next word in the given text. Instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words.

$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n) \quad \textbf{unigram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}) \quad \textbf{bigram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}w_{n-2}) \quad \textbf{trigram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}w_{n-2}w_{n-3}) \quad \textbf{4-gram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}w_{n-2}w_{n-3}w_{n-4}) \quad \textbf{5-gram}$$

- In general, **N-Gram** is

$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-N+1}^{n-1})$$

N-Grams
Computingprobabilitiesofwordsequences(Sentences)

Unigram
P(<s> the man from Jupiter came </s>)
P(the)P(man)P(from)P(jupiter)P(came)

Bigram
P(<s>theman fromJupitercame</s>)
P(the|<*s*>)P(man|the)P(from|man)P(jupiter|from)P(came|jupiter)P(</s>|came)

Trigram
P(<s>theman fromJupitercame</s>)
P(the|<*s*><*s*>)P(man|<*s*>the)P(from|theman)P(jupiter|manfrom)P(came|fromjupiter)
P(</s>|Jupiter came) P(</s>|came </s>)

N-GramsandMarkovModels

Theassumptionthattheprobabilityofaworddependsonlyonthepreviousword(s)iscalled **Markov assumption**.

Markovmodelsaretheclassofprobabilisticmodelsthatassumethatwecanpredictthe probability of some future unit without looking too far into the past.

•A**bigram**is called a **first-order**Markovmodel (because itlooksonetoken into the past)
•A**trigram**iscalleda**second-order**Markovmodel;
•Ingenerala**N-Gram**is calleda**N-1order**Markovmodel.

EstimatingN-GramProbabilities

Estimatingn-gramprobabilitiesiscalled**maximumlikelihoodestimation(**or**MLE**). WegettheMLEestimatefortheparametersofann-grammodelby***gettingcountsfroma corpus***, and **normalizing** the counts so that they lie between 0 and 1.

•**Estimatingbigramprobabilities:**

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

where C is the count of that pattern in the corpus

Estimating N-Gram probabilities

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

EstimatingN-GramProbabilities

ABigram Example

Amini-corpus:We augment eachsentence with a special symbol <s> atthe beginningofthesentence,togiveusthebigramcontextofthefirstword,and special end-symbol </s>.
<s>IamSam</s>
<s>SamIam</s>
<s>Ifly</s>

Uniquewords:I,am,Sam, fly

Bigrams:<s>and</s> arealsotokens.Thereare 6(4+2)tokens and 6*6=36 bigrams

P(I|<s>)=2/3	P(Sam|<s>)=1/3	P(am|<s>)=0	P(fly|<s>)=0	P(<s>|<s>)=0	P(</s>|<s>)=0
P(I|I)=0	P(Sam|I)=0	P(am|I)=2/3	P(fly|I)=1/3	P(<s>|I)=0	P(</s>|I)=0
P(I|am)=0	P(Sam|am)=1/2	P(am|am)=0	P(fly|am)=0	P(<s>|am)=0	P(</s>|am)=1/2
P(I|Sam)=1/2	P(Sam|Sam)=0	P(am|Sam)=0	P(fly|Sam)=0	P(<s>|Sam)=0	P(</s>|Sam)=1/2
P(I|fly)=0	P(Sam|fly)=0	P(am|fly)=0	P(fly|fly)=0	P(<s>|fly)=0	P(</s>|fly)=1
P(I|</s>)=0	P(Sam|</s>)=1/3	P(am|</s>)=1/3	P(fly|</s>)=1/3	P(<s>|</s>)=0	P(</s>|</s>)=0

Estimating N-Gram Probabilities

Example

<s> I am Sam </s>
<s> Sam I am </s>
<s> I fly </s>

Unigrams: I, am, Sam, fly

P(I)=3/8 P(am)=2/8 P(Sam)=2/8 P(fly)=1/8

Trigrams: There are 6*6*6=216 trigrams.

- Assume there are two tokens <s> <s> at the begining, and two tokens </s> </s> at the end.

P(I|<s> <s>)=2/3 P(Sam|<s> <s>)=1/3
P(am|<s> I)=1/2 P(fly|<s> I)=1/2
P(I|<s> Sam)=1
P(Sam|I am)=1/2 P(</s>|I am)=1/2
P(</s>|am Sam)=1
P(</s>|Sam </s>)=1

LanguagemodelEvaluation

Language model evaluation in Natural Language Processing (NLP) is a crucial step to assess the performance and quality of a language model. Evaluating language models helps researchersand developers understand how well a model performs on various tasks and benchmark it against other models or baselines. Here are some common evaluation techniques used in NLP:

1. **Perplexity:**Perplexityisawidelyusedmetricforevaluatinglanguagemodels. Itmeasureshowwell a language model predicts a given text corpus. Lower perplexity indicates better performance. Perplexity is calculated using the probabilities assigned by the model to each word in a test set.
2. **Accuracy:** Accuracy is a common evaluation metric for tasks like sentiment analysis, text classification, or named entity recognition. It measures the proportion of correctly predictedinstances compared to the total number of instances in a test set. Accuracy is usually calculated by comparing the predicted labels or outputs of the model with the ground truth labels.
3. **F1 Score:** F1 score is a widely used metric for evaluating models in tasks like named entity recognition,part-of-speechtagging,andinformationextraction.Itbalancesprecision(theproportion of correctly predicted positive instances) and recall (the proportion of actual positive instances correctly predicted). F1 score is the harmonic mean of precision and recall and provides a single measure of model performance.
4. **BLEU Score:** BLEU (Bilingual Evaluation Understudy) is a metric commonly used in machine translation to evaluate the quality of generated translations. It measures the overlap between the model's predicted translations and reference translations. BLEU score ranges from 0 to 1, with higher scores indicating better translation quality.
5. **ROUGE Score:** ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a family of metrics used to evaluate text summarization systems. ROUGE measures the overlap between the model's generated summaries and reference summaries. Different variants of ROUGE, such as ROUGE-N (measuring n-gram overlap) and ROUGE-L (measuring longest common subsequence), are used to evaluate different aspects of summarization quality.
6. **Human Evaluation:** In addition to automated metrics, human evaluation is often conducted to assess the quality of language models. Human evaluators provide judgments on various aspects of model outputs, such as fluency, coherence, grammaticality, relevance, and overall quality. Human evaluation provides valuable insights that may not be captured by automated metrics.

It is important to note that the choice of evaluation metric depends on the specific NLP task and the goals of the evaluation. Different metrics have their strengths and weaknesses, and a combination of metrics is often used to provide a comprehensive evaluation of a language model's performance.

ParameterEstimation

Parameter estimation in natural language processing (NLP) refers to the process ofestimatingthevaluesof modelparametersbased onobserveddata. In NLP, parameterestimationis crucial for various tasks, such as language modeling, machine translation, sentiment analysis,and part-of-speech tagging, among others.

TypesofParameterEstimations

1. Maximum-LikelihoodEstimationandSmoothing(MLE)
2. BayesianParameterEstimation
3. Large-ScaleLanguage Models

1.Maximum-LikelihoodEstimationandSmoothing (MLE)

Maximum Likelihood Estimation (MLE) is a common technique used in NaturalLanguage Processing (NLP) to estimate the parameters of a probabilistic language model. MLE aims to find the parameter values that maximize the likelihood of the observed data.

In the context of NLP, MLE is often used to estimate the probabilities of words or sequences of words in a language model. The basic idea is to count the occurrences of different words or sequences of words in a given training corpus and use these countsto calculate the probabilities as follows.

P(wi/wi-1,wi-2)=c(wi,wi-1,wi-2)/c(wi-1,wi-2)

Here'sa step-by-stepexplanationofhowMLEworksinNLP:

1. **Data Collection:** Collect a large corpus of text that represents the domain or language you want to model. This corpus will be used as the training data.
2. **Data Preprocessing:** Preprocess the training data by tokenizing it into words or subword units, removing punctuation, normalizing case, etc. This step ensures that the data is in a suitable format for modeling.
3. **Parameter Estimation:** Calculate the probabilities of words or sequences of words basedonthetrainingdata.Forexample,toestimate theprobabilityofaword,countthe number of occurrences of that word in the corpus and divide it by the total number of words in the corpus.
4. **Smoothing:** One challenge in using MLE is that unseen words or sequences in the training data will have zero probabilities. To address this issue, smoothing techniquesare often applied to assign non-zero probabilities to unseen events. Commonsmoothing methods include Laplace smoothing (add-one smoothing), Lidstone smoothing, and Good-Turing smoothing.

5.**Model Evaluation:** Evaluate the performance of the language model using evaluation metrics such as perplexity, accuracy, or F1 score. These metrics help assess how well the model predicts the target language or performs on specific NLP tasks.

MLE is a fundamental approach for estimating language model probabilities in NLP. However,ithassomelimitations.MLE assumesthatthetrainingdataisrepresentativeof the target distribution and that each data point is independent. In practice, these assumptions may not always hold, leading to potential limitations in the performance of the language model. Researchers and practitioners often explore more sophisticated techniques, such asneural language models, to overcome theselimitationsand improve language modeling in NLP.

2. BayesianParameterEstimation

Bayesian parameter estimation is an alternative approach to estimating the parameters of a language model in Natural Language Processing (NLP). Unlike Maximum LikelihoodEstimation (MLE), which provides point estimates of the parameters, Bayesian estimation incorporates prior knowledge and uncertainty into the parameter estimation process.

In Bayesian parameter estimation, the goal is to estimate the posterior distribution of the parameters given the observed data and prior knowledge. The posterior distribution representsthe updated belief about the parameter values after considering the observed data. This distribution is obtained by combining the likelihood of the data and the prior distribution of the parameters using Bayes' theorem.

Here's astep-by-step explanationofhowBayesian parameterestimationworksinNLP:

1. Define a Prior Distribution: Specify a prior distribution that captures the initial belief or knowledge about the parameters before observing any data. The prior distribution represents the uncertainty or prior information about the parameter values.
2. Likelihood Calculation: Calculate the likelihood of the observed data given the parameters. The likelihood measures how probable the observed data is under the current parameter values.
3. Bayesian Inference: Apply Bayes' theorem to update the prior distribution based on thelikelihood and obtain the posterior distribution of the parameters. The posterior distribution reflects the updated belief about the parameter values after considering the observed data.
4. Posterior Analysis: Analyze the posterior distribution to obtain estimates of the parameters. This can be done by calculating the mean, median, or mode of the posterior distribution, which represent the point estimates of the parameters. Additionally, credible intervals can be computed to quantify the uncertainty in the parameter estimates.
5. Model Evaluation: Evaluate the performance of the Bayesian language model using evaluation metrics such as perplexity, accuracy, or F1 score, similar to the evaluation of MLE-basedmodels.

Bayesian parameter estimation provides several advantages over MLE in NLP. It allows the incorporation of prior knowledge, which can be particularlyuseful when the amount of available training data is limited. Additionally, Bayesian estimation provides a richer representation of uncertainty by yielding a posterior distribution rather than a point estimate. This uncertainty information can be valuable for decision-making or downstream applications.

However, Bayesian parameter estimation can be computationally demanding, especially for complex models with high-dimensional parameter spaces. Techniques like Markov Chain Monte Carlo (MCMC) or variational inference are commonly used to approximate the posterior distribution when direct computation is infeasible.

Overall, Bayesianparameterestimationoffers aprincipledandflexibleframeworkforestimating parameters in language models, incorporating prior knowledge, and capturing uncertainty inNLP tasks.

LanguageModelAdaption

Language model adaptation in NLP refers to the process of fine-tuning a pre-trained language model on a specific task or domain to improve its performance. Language models like GPT-3.5 are trained on a large corpus of diverse text from the internet, which gives them a general understanding of language. However, fine-tuning or adapting them to a specific task or domain can enhance their performance on that particular task.

Herearethe generalsteps involvedinlanguagemodel adaptation:

1. **Dataset Collection:** Gather a dataset that is specific to the target task or domain you want to adapt the language model to. This dataset should be representative of the target task and should include text samples that the language model is likely to encounter during inference.
2. **Dataset Preprocessing:** Preprocess the collected dataset by cleaning the text, removing irrelevant information, and ensuring the dataset is in a format that can be used for training.
3. **Model Architecture:** Decide on the specific architecture you want to use for adaptation. This can be the same architecture as the pre-trained language model or a modified version depending on the task requirements.
4. **Task-SpecificLabels:** If yourtargettaskinvolves supervisedlearning, you willneedtoannotate or label your dataset with the appropriate task-specific labels. This step is crucial for tasks like sentiment analysis, named entity recognition, or machine translation.
5. **Fine-tuning:** Initialize the pre-trained language model with the weights learned during pre-training, and then fine-tune it on your target dataset. During fine-tuning, the model learns task-specific patterns and representations. The process typically involves minimizing a task-specific loss function using techniques such as backpropagation and gradient descent.
6. **Hyperparameter Tuning:** Experiment with different hyperparameters such as learning rate, batch size, and regularization techniques to optimize the performance of the adapted model. This step helps to find the best set of hyperparameters that work well for the specific task.
7. **Evaluation and Iteration:** Evaluate the performance of the adapted language model on a validationsetorusingotherevaluationmetricsspecifictoyourtask.Iterateonthefine-tuning

processbymakingchangestothearchitecture,hyperparameters,ordataset,ifnecessary,to further improve performance.

It's worth noting that language model adaptation requires a substantial amount of task-specific data to achieve good performance. If onlya limited amount of data is available, transfer learning techniques such as few-shot or zero-shot learning can be used to leverage the knowledge learned from the pre-trained language model. Additionally, it's important to strike a balance between task-specific adaptation and preserving the general language understanding of the original pre-trained model.

TypesofLanguageModels

Therearevarioustypesoflanguagemodelsusedinnaturallanguageprocessing(NLP)thatdiffer in their architecture, training methods, and intended use cases. Here are some commonly used types of language models in NLP:

1. **N-gram Language Models**: N-gram models are simple and widely used language models that predict the probabilityof a word given its preceding context of n-1 words. Theyrelyon counting and statistical techniques to estimate the probabilities. For example, a trigram model predicts the next word based on the previous two words.
2. **Neural Language Models**: Neural language models leverage deep learning techniques, such as recurrentneuralnetworks(RNNs),longshort-termmemory(LSTM)networks,andtransformers, to capture complex patterns and dependencies in language. These models learn distributed representations of words and generate probabilities based on the context.
3. **Transformer-based Language Models**: Transformer models, such as OpenAI's GPT (Generative Pre-trained Transformer), have gained significant attention in recent years. These models use self-attention mechanisms to capture global dependencies and parallelize computation, making them highly effective for tasks like language generation, machine translation, and question answering.
4. **Contextualized Language Models**: Contextualized language models, such as ELMo (Embeddings from Language Models) and BERT (Bidirectional Encoder Representations from Transformers), generate word representations that are sensitive to the context in which the word appears. These models provide contextual embeddings that areuseful for downstream NLP tasks like sentiment analysis, named entity recognition, and text classification.
5. **Encoder-Decoder Models**: Encoder-decoder models, often based on the sequence-to-sequence architecture, are used for tasks like machine translation and text summarization. These models consist of an encoder that processes the input sequence and a decoder that generates the output sequence based on the encoded representation.
6. **Hybrid Models**: Some language models combine multiple approaches or architectures to leverage their individual strengths. For example, ULMFiT (Universal Language Model Fine-tuning) combines features of transformer models with techniques like transfer learning and fine-tuning to improve performance on specific tasks.
7. **Pre-trained Language Models**: Pre-trained language models, such as GPT, BERT, and RoBERTa, are models that are trained on large amounts of data before being fine-tuned for specific tasks. These models capture a broad understanding of language and can be adapted to various downstream tasks with additional training.

These are just a few examples of the types of language models used in NLP. Different models have different strengths and are suited for specific tasks and scenarios. Researchers continue to explore new architectures and techniques to develop more powerful and efficient language models for a wide range of NLP applications.

Language-SpecificModellingProblems

In natural language processing (NLP), there are several language-specific modeling problemsthat arise due to the unique characteristics and complexities of different languages. Here aresome common language-specific modeling challenges in NLP:

1. **Morphological Variations:** Many languages exhibit rich morphology, where words can have multiple forms depending on factors like tense, number, gender, and case. Modeling morphological variations can be challenging, especially for languages with highly inflected forms,asitrequires capturingtheintricaterelationshipsbetweenword formsandtheirmeanings.
2. **Word Order and Syntax:** Languages vary in their word order and sentence structure. For example, English follows a subject-verb-object (SVO) order, while languages like Japanese follow a subject-object-verb (SOV) order. Modeling the syntactic structure and understandingthe correct word order for different languages is crucial for tasks like parsing, machine translation, and text generation.
3. **Named Entity Recognition (NER):** Named Entity Recognition involves identifying and classifying named entities such as names of people, organizations, locations, and dates in text. Different languages have specific naming conventions and variations, which pose challenges for building language-independent NER systems. Language-specific rules and resources are often required to handle these variations effectively.
4. **SentimentAnalysis:** Sentimentanalysisaimstodeterminethesentimentoropinionexpressedin a text. The sentiment lexicons, patterns, and linguistic cues that indicate sentiment can differ across languages. Building accurate sentiment analysis models requires language-specific sentiment resources and training data that capture the nuances of sentiment expression in different languages.
5. **Language Idiosyncrasies:** Each language has its own set of idiosyncrasies, such as idiomatic expressions, proverbs, and cultural references. These language-specific nuances pose challenges formodelsthat aimtounderstand and generatenaturallanguagetext. Capturingand representing these idiosyncrasies is crucial for building language models that accurately handle language-specific contexts.
6. **Low-Resource Languages:** Low-resource languages, which have limited amounts of training data and linguistic resources, present unique modeling challenges. Limited resources make it difficult to develop robust models for these languages. Techniques like cross-lingual transfer learning, data augmentation, and unsupervised methods are often employed to address the scarcity of language-specific resources.
7. **Code-SwitchingandMultilingualism:** Manylanguagesexhibitcode-switching,wherespeakers alternate between two or more languages within a conversation. Modeling code-switching and multilingual text requires techniques that can handle language mixing, language identification, and understanding the context-dependent language usage.

Addressing these language-specific modeling challenges in NLP often involves a combination of linguistic knowledge, domain expertise, and language-specific resources. Researchers and practitioners work on developing techniques that can effectively handle these challenges and improve the performance and applicability of NLP models across different languages.

MultilingualandCrosslingualLanguageModelling.

Multilingual and cross-lingual language modeling in NLP involves developing models that can understand, generate, or process multiple languages. Here's an overview of multilingual and cross-lingual language modeling:

1. **Multilingual Language Modeling:** Multilingual language models are trained to handle multiple languages simultaneously. These models are typically trained on a diverse multilingual corpus and aim to capture shared linguistic properties across languages. By leveraging the similarities between languages, multilingual models can generalize well across different language contexts. They can be used for tasks like text classification, named entity recognition, and sentiment analysis in multiple languages.
2. **Cross-lingual Language Modeling:** Cross-lingual language models focus on bridging the gap between different languages. These models are trained on a source language and then applied to a target language, even if the target language has little or no labeled data. Cross-lingual models enable transfer learning, where knowledge learned from a resource-rich language can be transferred to improve performance in a low-resource language. This approach is particularly useful for tasks like machine translation, cross-lingual document retrieval, and cross-lingual information extraction.
3. **Bilingual and Multilingual Embeddings:** Bilingual and multilingual word embeddings represent words from different languages in a shared vector space. These embeddings capture semantic and syntactic similarities between words in multiple languages. Bilingual embeddings map words from one language to their counterparts in another language, while multilingual embeddings aim to represent words from multiple languages in a single vector space. These embeddings facilitate cross-lingual tasks by allowing for direct comparisons and knowledge transfer across languages.
4. **MachineTranslation:**Machinetranslationisaclassicexampleofcross-linguallanguage modeling. Neural machine translation models, such as sequence-to-sequence models, use an encoder-decoder architecture to translate text from one language to another. These models can be trained with parallel corpora containing source-target language pairs, and they learn to map the source language to the target language. Transfer learning techniques and multilingual training can improve the translation performance across multiple languages.
5. **Zero-ShotandFew-ShotLearning:**Zero-shotandfew-shotlearningtechniquesenable cross-lingual transfer without requiring labeled data in thetarget language. In zero-shot

learning, a model can generate outputs in a language it has not been explicitly trained on, using only high-level instructions or prompts. Few-shot learning involves training a model on a limited amount of labeled data from the target language, allowing it to performwellonthatlanguage.Thesetechniquesareusefulwhenresourcesinthetarget language are scarce.

6. **Cross-lingual Named Entity Recognition (NER):** Cross-lingual NER involvesrecognizing named entities in multiple languages. By leveraging multilingual training data and shared representations, cross-lingual NER models can generalize across languagesandrecognizeentitieseveninlow-resourcelanguages.Techniqueslikecross-lingual transfer learning and alignment methods are used to improve cross-lingual NER performance.

Multilingual and cross-lingual language modeling techniques have contributed significantlytomakingNLPmoreaccessibleandeffectiveacrossdifferentlanguagesand language pairs. They enable the development of models that can handle diverse linguistic contexts, promote knowledge transfer, and facilitate communication and understanding in a multilingual world.