Unit-V

Language Modeling

- Introduction
- n-Gram Models
- Language Model Evaluation
- Parameter Estimation
- Language Model Adaptation
- Types of Language Models
- Language Specific Modeling Problems
- Multilingual and Cross-lingual Language Modeling

Introduction

- Statistical Language Model is a model that specifies the a priori probability of a particular word sequence in the language of interest.
- Given an alphabet or inventory of units ∑ and a sequence W= w1w2…..wt ϵ ∑* a language model can be used to compute the probability of W based on parameters previously estimated from a training set.
- The inventory ∑ is the list of unique words encountered in the training data.
- Selecting the units over which a language model should be defined is a difficult problem particularly in languages other than English.

Introduction

- A language model is combined with other model or models that hypothesize possible word sequences.
- In speech recognition a speech recognizer combines acoustic model scores with language model scores to decode spoken word sequences from an acoustic signal.
- Language models have also become a standard tool in information retrieval, authorship identification, and document classification.

n-Gram Models

- Finding the probability of a word sequence of arbitrary length is not possible in natural language because natural language permits infinite number of word sequences of variable length.
- The probability P(W) can be decomposed into a product of component probabilities according to the chain rule of probability:

$$P(W) = P(w_1 \ldots w_t) = P(w_1) \prod_{i=1}^{t} P(w_i | w_{i-1} w_{i-2} \ldots w_2 w_1)$$

- Since the individual terms in the above product are difficult to compute directly n-gram approximation was introduced.

n-Gram Models

- The assumption is that all the preceding words except the n-1 words directly preceding the current word are irrelevant for predicting the current word.
- Hence P(W) is approximated to:

$$P(W) \approx \prod_{i=1}^{t} P(w_i | w_{i-1}, \dots w_{i-n+1})$$

- This model is also called as (n-1)-th order Markov model because of the assumption of the independence of the current word given all the words except for the n-1 preceding words.

Language Model Evaluation

- Now let us look at the problem of judging the performance of a language model.
- The question is how can we tell whether the language model is successful at estimating the word sequence probabilities?
- Two criteria are used:
- Coverage rate and perplexity on a held out test set that does not form part of the training data.
- The coverage rate measures the percentage of n-grams in the test set that are represented in the language model.
- A special case is the out-of-vocabulary rate (OOV) which is the percentage of unique word types not covered by the language model.

Language Model Evaluation

- The second criterion perplexity is an information theoretic measure.
- Given a model p of a discrete probability distribution, perplexity can be defined as 2 raised to the entropy of p:

$$PPL(p) = 2^{H(p)} = 2^{-\sum_x p(x) log_2 p(x)}$$

- In language modeling we are more interested in the performance of a language model q on a test set of a fixed size, say t words (w1w2wt).
- The language model perplexity can be computed as:
- q(wi) computes the probability of the ith word.

$$2^{-\frac{1}{t}\sum_{i=1}^{t} log_2 q(w_i)}$$

$$PPL(p,q) = 2^{H(p,q)} = 2^{-\sum_{i=1}^{t} p(w_i) log_2 q(w_i)}$$

Language Model Evaluation

- If q(wi) is an n-gram probability, the equation becomes

$$2^{-\frac{1}{l}\sum_{i=1}^{l} log_2 p(w_i | w_{i-1}, \ldots, w_{i-n+1})}$$

- When comparing different language models, their perplexities must be normalized with respect to the same number of units in order to obtain a meaningful comparison.
- Perplexity is the average number of equally likely successor words when transitioning from one position in the word string to the next.
- If the model has no predictive power, perplexity is equal to the vocabulary size.

Language Model Evaluation

- A model achieving perfect prediction has a perplexity of one.
- The goal in language model development is to minimize the perplexity on a held-out data set representative of the domain of interest.
- Sometimes the goal of language modeling might be to distinguish between “good” and “bad” word sequences.
- Optimization in such cases may not be minimizing the perplexity.

Parameter Estimation

- Maximum-Likelihood Estimation and Smoothing
- Bayesian Parameter Estimation
- Large-Scale Language Models

Maximum-Likelihood Estimation and Smoothing

- The standard procedure in training n-gram models is to estimate n-gram probabilities using the maximum-likelihood criterion in combination with parameter smoothing.
- The maximum-likelihood estimate is obtained by simply computing relative frequencies:

$$P(w_i|w_{i-1}, w_{i-2}) = \frac{c(w_i, w_{i-1}, w_{i-2})}{c(w_{i-1}, w_{i-2})}$$

- Where c(wi,wi-1,wi-2) is the count of the trigram wi-2wi-1wi in the training data.

Maximum-Likelihood Estimation and Smoothing

- This method fails to assign nonzero probabilities to word sequences that have not been observed in the training data.
- The probability of sequences that were observed might also be overestimated.
- The process of redistributing probability mass such that peaks in the n-gram probability distribution are flattened and zero estimates are floored to some small nonzero value is called smoothing.
- The most common smoothing technique is **backoff**.

Maximum-Likelihood Estimation and Smoothing

- Backoff involves splitting n-grams into those whose counts in the training data fall below a predetermined threshold Ʈ and those whose counts exceed the threshold.
- In the former case the maximum-likelihood estimate of the n-gram probability is replaced with an estimate derived from the probability of the lower-order (n-1)-gram and a backoff weight.
- In the later case, n-grams retain their maximum-likelihood estimates, discounted by a factor that redistributes probability mass to the lower-order distribution.

Maximum-Likelihood Estimation and Smoothing

- The back-off probability P_{BO} for wi given wi-1,wi-2 is computed as follows:

$$P_{BO}(w_i|w_{i-1},w_{i-2}) = \begin{cases} d_c P(w_i|w_{i-1},w_{i-2}) \text{ if } c > \tau \\ \alpha(w_{i-1},w_{i-2}) P_{BO}(w_i|w_{i-1}) \text{otherwise} \end{cases}$$

- Where c is the count of (wi,wi-1,wi-2), and dc is a discounting factor that is applied to the higher order distribution.
- The normalization factor α(wi-1,wi-2) ensures that the entire distribution sums to one and is computed as:

$$\alpha(w_{i-1},w_{i-2}) = \frac{1 - \sum_{w_i:c(w_i,w_{i-1},w_{i-2})>\tau} d_c P(w_i|w_{i-1},w_{i-2})}{\sum_{w_i:c(w_i,w_{i-1},w_{i-2})\leq\tau} P_{BO}(w_i|w_{i-1})}$$

Maximum-Likelihood Estimation and Smoothing

- The way in which the discounting factor is computed determines the precise smoothing technique.
- Well-known techniques include:
  - Good-Turing
  - Written-Bell
  - Kneser-Ney
- In Kneser-Ney smoothing a fixed discounting parameter D is applied to the raw n-gram counts before computing the probability estimates:

$$P_{KN}(w_i|w_{i-1}, w_{i-2}) = \begin{cases} \frac{max\{c(w_i,w_{i-1},w_{i-2})-D,0\}}{\sum_{w_i} c(w_i,w_{i-1},w_{i-2})} & \text{if } c > \tau \\ \alpha(w_{i-1}, w_{i-2}) P_{KN}(w_i|w_{i-1}) & \text{otherwise} \end{cases}$$

Maximum-Likelihood Estimation and Smoothing

- In modified Kneser-Ney smoothing, which is one of the most widely used techniques, different discounting factors D1,D2,D3+ are used for n-grams with exactly one, two, or three or more counts:

$$Y = \frac{n_1}{n_1 + 2 * n_2}$$

$$D_1 = 1 - 2Y\frac{n_2}{n_1}$$

$$D_2 = 2 - 3Y\frac{n_3}{n_2}$$

$$D_{3+} = 3 - 4Y\frac{n_4}{n_3}$$

- Where n1,n2,….. are the counts of n-grams with one, two, …, counts.

Maximum-Likelihood Estimation and Smoothing

- Another common way of smoothing language model estimates is linear model interpolation.
- In linear interpolation, M models are combined by

$$P(w_i|w_{i-1}, w_{i-2}) = \sum_{m=1}^{M} \lambda_m P(w_i|h_m)$$

- Where λ is a model-specific weight.
- The following constraints hold for the model weights: 0<= λ<=1 and ∑m λm =1.
- Weights are estimated by maximizing the log-likelihood on a held-out data set that is different from the training set for the component models.

Maximum-Likelihood Estimation and Smoothing

- This is done using the expectation-maximization (EM) procedure.

Bayesian Parameter Estimation

- This is an alternative parameter estimation method where the set of parameters are viewed as a random variable governed by a prior statistical distribution.
- Given a training sample S and a set of parameters $\theta, P(\theta)$ denotes a prior distribution over different possible values of θ, and P(θ/S) is the posterior distribution and is expressed using Baye's rule as:

$$P(\theta|S) = \frac{P(S|\theta)P(\theta)}{P(S)}$$

Bayesian Parameter Estimation

- In language modeling, $\theta = < P(w1), \ldots., P(Wk) >$ (where K is the vocabulary size) for a unigram model.
- For an n-gram model θ=<P(W1/h1),…,P(Wk/hk)> with K n-grams and history h of a specified length.
- The training sample S is a sequence of words, W1…..Wt.
- We require a point estimate of θ given the constraints expressed by the prior distribution and the training sample.
- A maximum a posterior (MAP) can be used to do this.

$$\theta^{MAP} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, P(\theta|S) = \underset{\theta \in \Theta}{\operatorname{argmax}}\, P(S|\theta)P(\theta)$$

Bayesian Parameter Estimation

- The Bayesian criterion finds the expected value of θ given the sample S:

$$\theta^B = E[\theta|S] = \int_\Theta \theta P(\theta|S) d\theta$$

$$= \frac{\int_\Theta \theta P(S|\theta)P(\theta)d\theta}{\int_\Theta P(S|\theta)P(\theta)d\theta}$$

- Assuming that the prior distribution is a uniform distribution, the MAP is equivalent to the maximum-likelihood estimate.

Bayesian Parameter Estimation

- Bayesian estimate is equivalent to the maximum-likelihood estimate with Laplace smoothing:

$$\theta_w^B = \frac{c(w) + 1}{\sum_w c(w) + K}$$

- Different choices for the prior distribution lead to different estimation functions.
- The most commonly used prior distribution in language model is the Dirichlet distribution.

Bayesian Parameter Estimation

- The Dirichlet distribution is the conjugate prior to the multinomial distribution. It is defined as:

$$p(\theta) = D(\alpha_1, \ldots, \alpha_K) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

- Where Γ is the gamma function and α1,….. Ak are the parameters of the Dirichlet distribution.
- It can also be thought of as counts derived from an a priori training sample.

Bayesian Parameter Estimation

- The MAP estimate under the Dirichlet prior is:

$$\theta^{MAP} = \underset{\theta \in \Theta}{\operatorname{argmax}} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{n_k + \alpha_k - 1}$$

- Where nk is the number of times word k occurs in the training sample.
- The result is another Dirichlet distribution parameterized by nk+ α
- The MAP estimate of P(θ/W,α) thus is equivalent to the maximum-likelihood estimate with add-m smoothing.
- mk= αk-1 that is pseudocounts of size αk-1 are added to each word count.

Large-Scale Language Models

- As the amount of available monolingual data increases daily models can be built from sets as large as several billions or trillions of words.
- Scaling language models to data sets of this size requires modifications to the ways in which language models are trained.
- There are several approaches to large-scale language modeling.
- The entire language model training data is subdivided into several partitions, and counts or probabilities derived from each partition are stored in separate physical locations.
- Distributed language modeling scales to vary large amounts of data and large vocabulary sizes and allows new data to be added dynamically without having to recompute static model parameters.

Large-Scale Language Models

- The drawback of distributed approaches is the slow speed of networked queries.
- One technique uses raw relative frequency estimate instead of a discounted probability if the n-gram count exceeds the minimum threshold (in this case 0):

$$S(w_i|w_{i-1}, w_{i-2}) = \begin{cases} P(w_i|w_{i-1}, w_{i-2}) \text{ if } c > 0 \\ \alpha S(w_i|w_{i-1}) \text{ otherwise} \end{cases}$$

- The α parameter is fixed for all contexts rather than being dependent on the lower-order n-gram.

Large-Scale Language Models

- An alternative possibility is to use large-scale distributed language models at a second pass rescoring stage only, after first-pass hypotheses have been generated using a smaller language model.
- The overall trend in large-scale language modeling is to abandon exact parameter estimation of the type described in favor of approximate techniques.

Language Model Adaptation

- Language model adaptation is about designing and tuning model such that it performs well on a new test set for which little equivalent training data is available.
- The most commonly used adaptation method is that of mixture language models or model interpolation.
- One popular method is topic-dependent language model adaptation.
- The documents are first clustered into a large number of different topics and individual language models can be built for each topic cluster.
- The desired final model is then fine-tuned by choosing and interpolating a smaller number of topic-specific language models.

Language Model Adaptation

- A form of dynamic self-adaptation of a language model is provided by trigger models.
- The idea is that in accordance with the underlying topic of the text, certain word combinations are more likely than other to co-occur.
- Some words are said to trigger others for example the words stock and market in a financial news text.