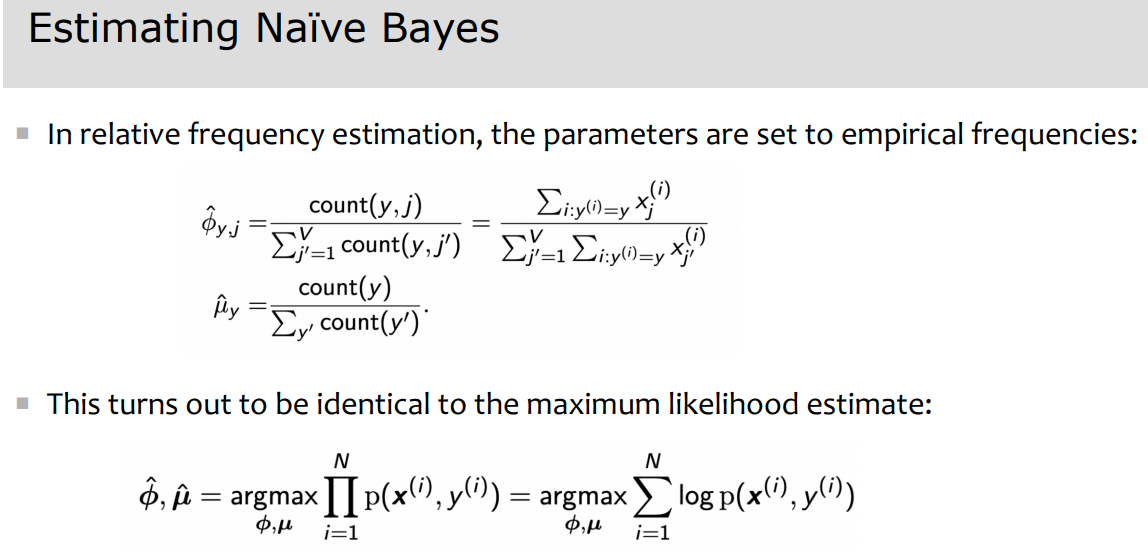
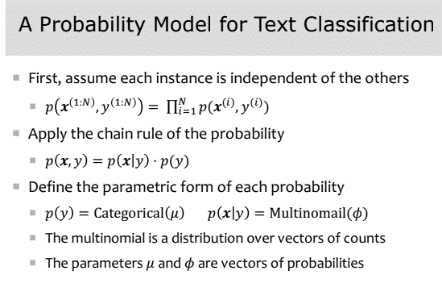
number of ways to select k words out of n given words (“unordered samples without replacement”)

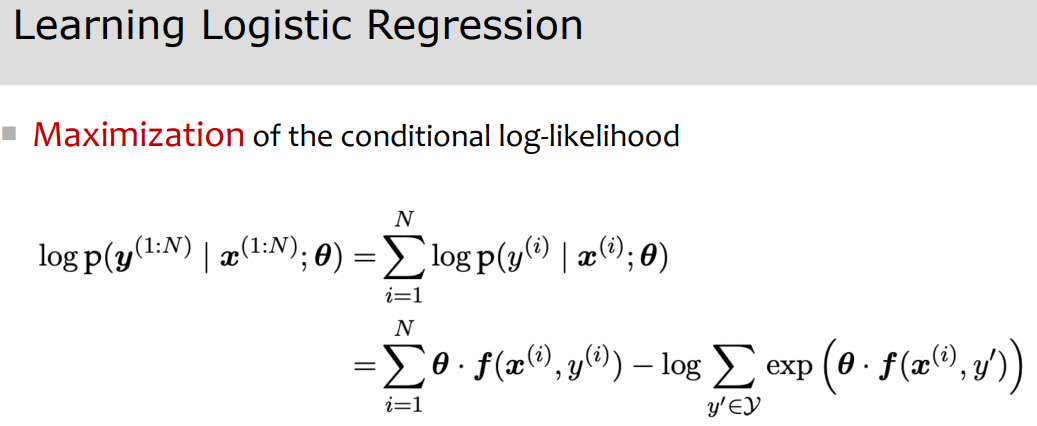
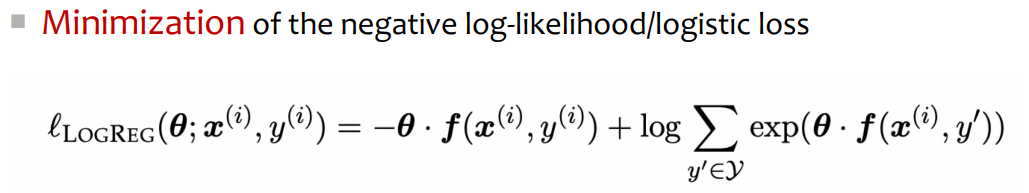
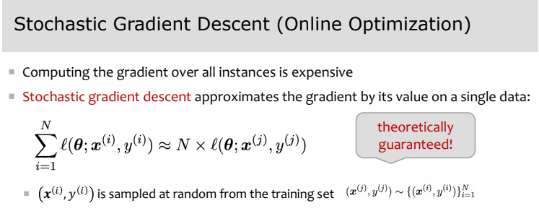
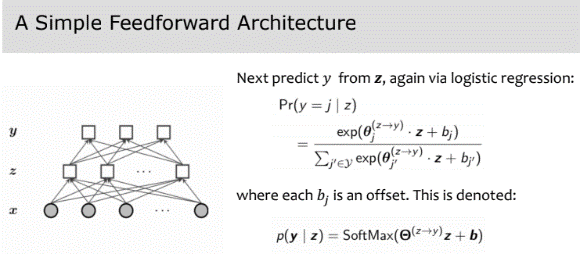
 

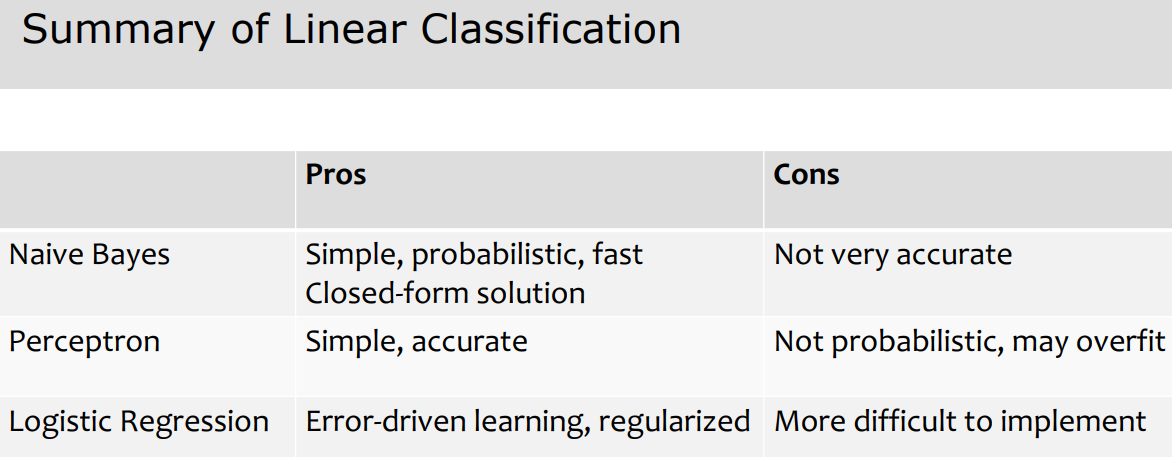
Perception classification is **discriminative**

Naïve Bayes is **probabilistic**

Logistic Regression is both **discriminative and probabilistic** (it computes the conditional probability of the label)

We estimate the parameters of Naïve Bayes through **maximum likelihood** (maximizing likelihood of dataset)



 Softmax + negative LL loss = cross-entropy loss in deep learning

Evaluation metrics:

1. Accuracy (problem = doesn’t take into account precision/recall for unbalanced datasets)
2. Recall = TP / (TP + FN) (fraction of positive instances which were correctly classified)
3. Precision = TP / (TP + FP) (fraction of positive predictions that were correct)
4. F1 score = 2\*recall\*precision / (recall + precision) (tradeoff between just recall or just precision)
5. Recall/precision imply binary classif.. In mcc, instances are positive for one class and negative for other classes
6. Two ways to combine performance across classes:
   1. Macro F-measure: compute F-score per class, and average across all classes (treats all classes equally)
   2. Micro F-measure: compute the total number of TP, FP, and FN across all classes, and compute a single F-score. This emphasizes performance on high-frequency classes

Markov processes: have sequence of N r.v.s, want a sequence probability model. There are |V|^n possible sequences

First order markov process: P(x1, x2, …, xn) = P(x1)P(xi | xi-1)

Second order markov process: P(x1, x2, …, xn) = P(x1)P(x2 | x1)P(xi | xi-1, xi-2)

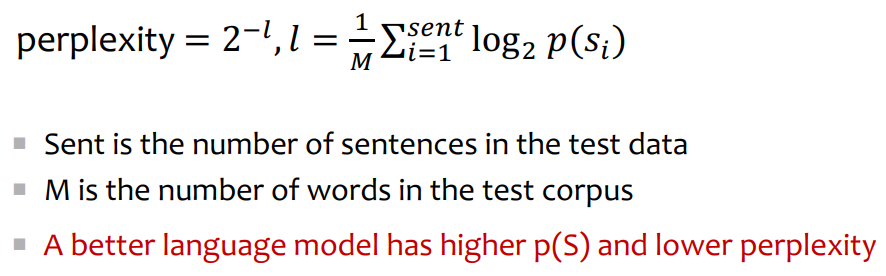
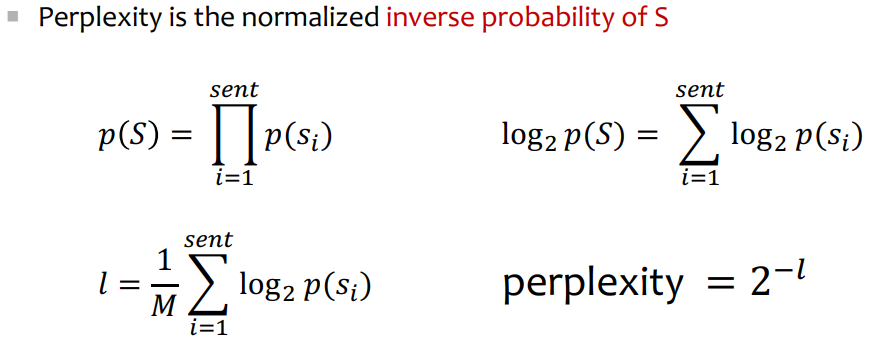
How to evaluate a language model:

1. Extrinsic: build a new language model, use it for some task (speech recognition, machine translation)

**Time-consumin**g and **Bad approximation** unless test data looks like training data (so, only useful in pilot experiments)

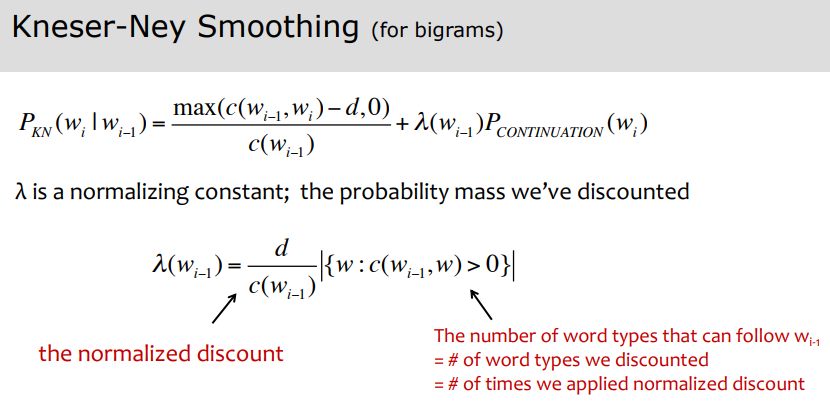
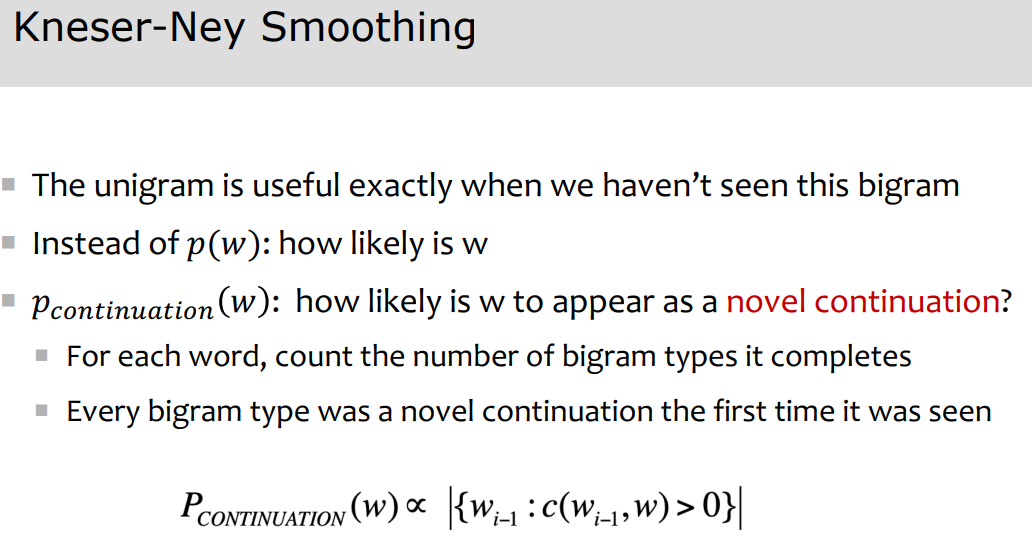
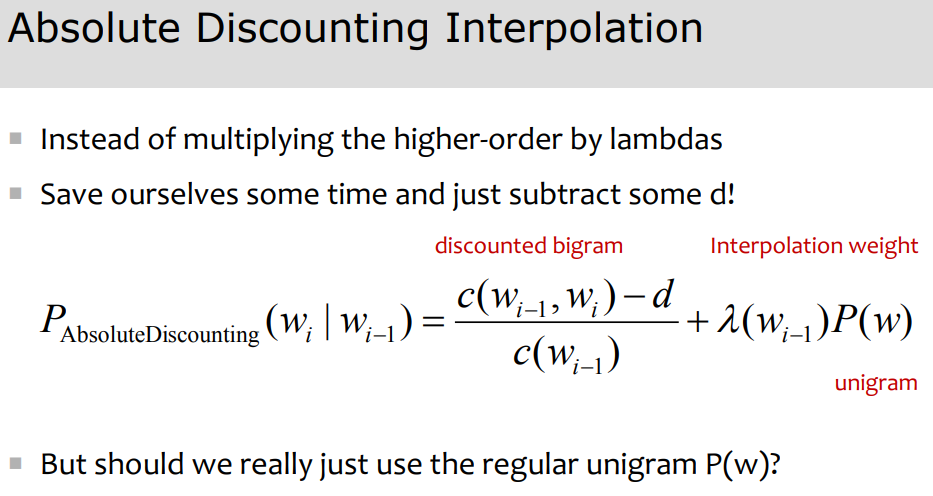
1. Intrinsic: measure how good we are at modeling language
   1. Perplexity (Cannot compare perplexities of language models trained on different corpora)

**Problem with add one smoothing:** with a large vocabulary, it thinks we are extremely likely to see novel events, rather than words we’ve actually seen (a large dictionary makes novel events too probable)

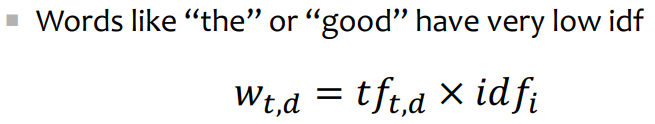
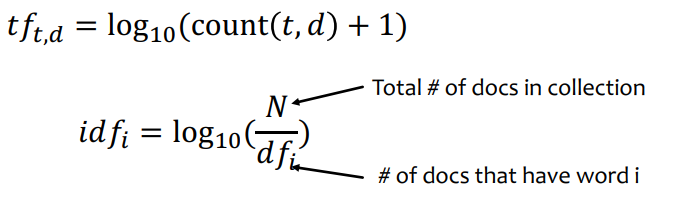
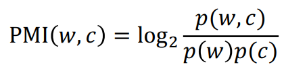


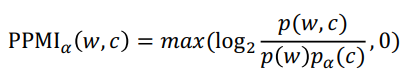
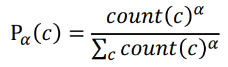
**Interpolation –** mixture of unigram, bigram, trigram, etc. models

**Backoff –** use trigram if good evidence, otherwise bigram, otherwise unigram



TFIDF (term-frequency inverse document frequency):

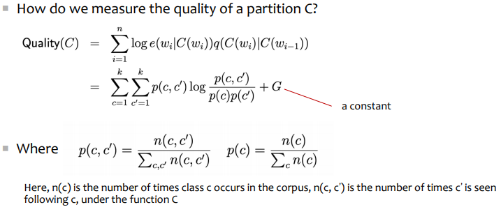


PMI (Pointwise Mutual Information):

Do word w and c co-occur more than if they were independent?

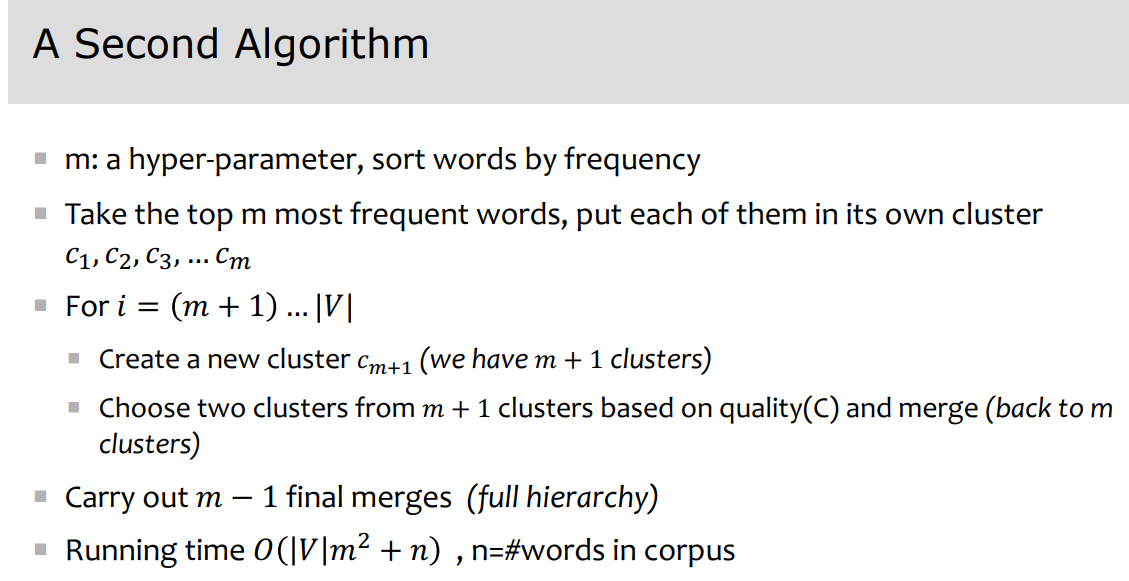
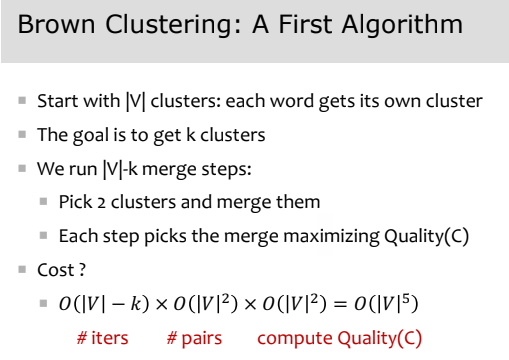
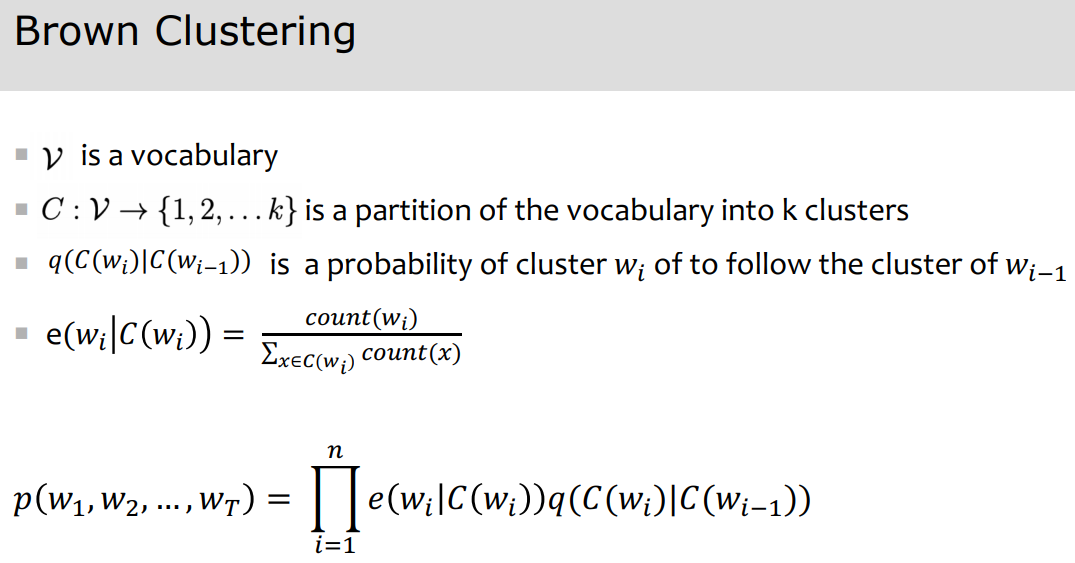
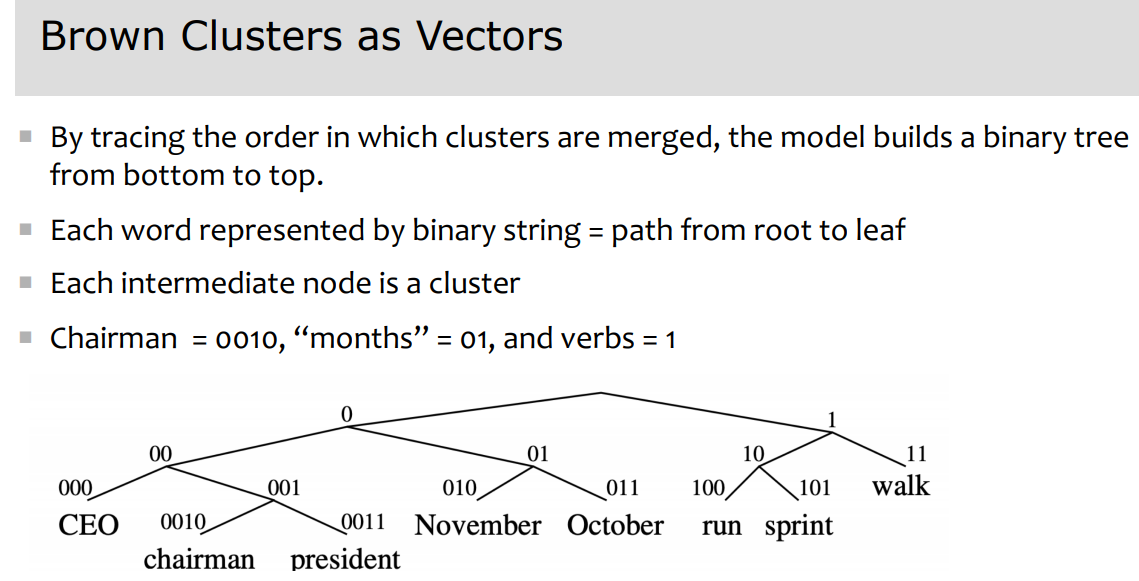
PPMI (Positive Pointwise Mutual Information):

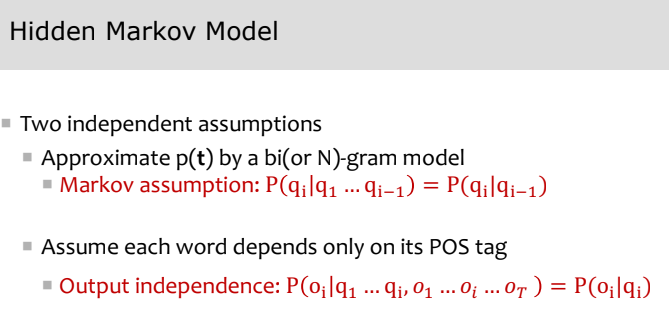
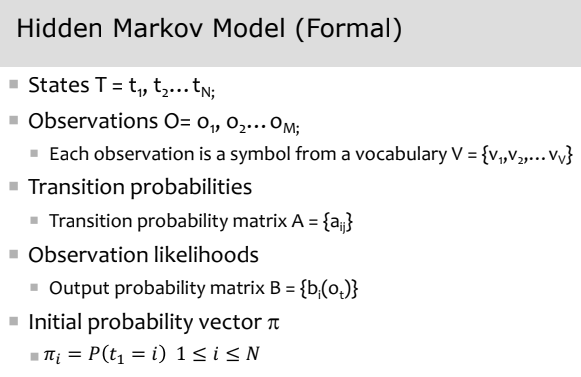
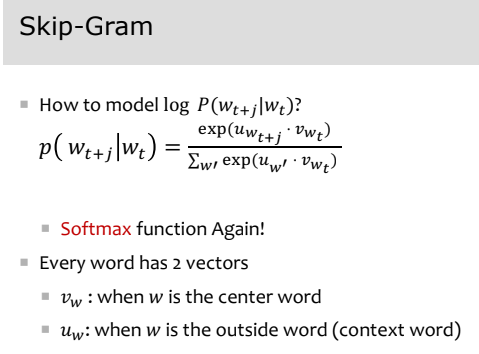
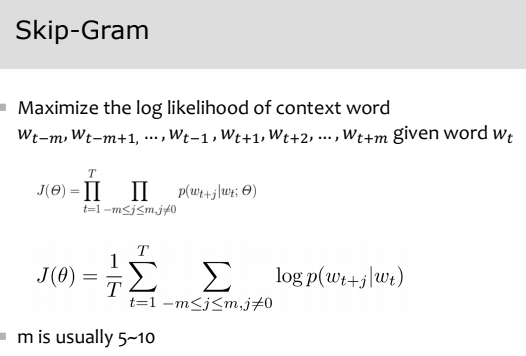
- **PMI is biased toward infrequent events**. Very rare words have very high PMI values. We want to give rare words

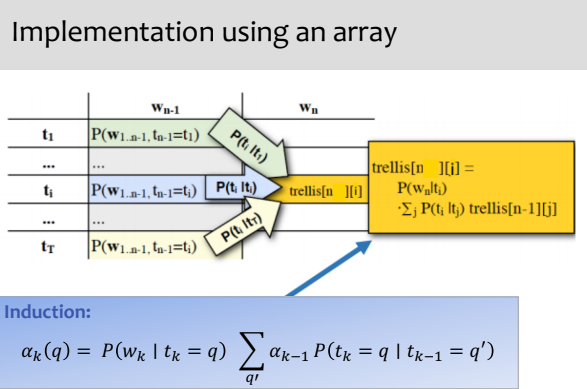
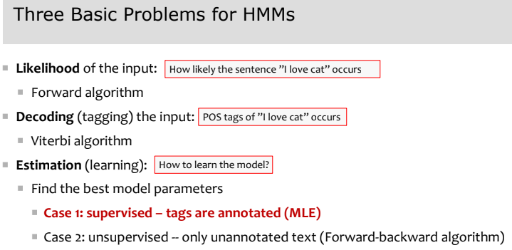
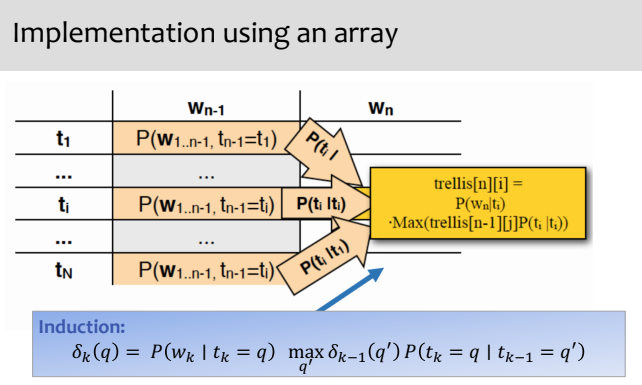
slightly higher probabilities. **PPMI vectors are long and sparse.** Want to learn vectors that are short and dense (because easier to use, generalize better than storing counts, and in practice work better)

How to get short, dense vectors?

1. SVD (a special case of this is called LSA – Latent Semantic Analysis)
2. Brown clustering
3. Neural language modeling (skip-grams and CBOW – Continuous bag-of-words)





 🡨 Forward Algorithm ; Viterbi Algorithm 🡪

**Context-free grammar -** A CFG gives a formal way to define what meaningful constituents are and exactly how a constituent is formed out of other constituents (or words). It defines valid structure in a language.

