**Guest Lecture**

Representation Learning for Grounded Special Reasoning – Michael Jenner

**Lecture 2 – Topic Models**

Topic for today: Language Models (LM)

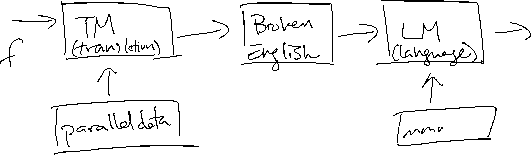
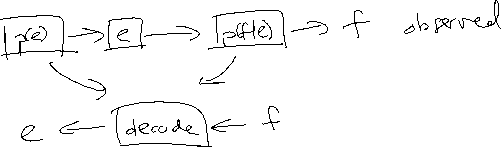
1. Applications of LM
   1. Noisy-channel models
2. Definition of N-gram models (particular type of LM)
   1. Decompositionality
3. Evaluation: Perplexity
4. Smoothing

**Activity: Guess the Author (refer to slides)**

* Someone guesses machine correctly
* Output was from model trained with Shakespearean text

**Definition of LM**

* Assume we are given a finite vocabulary
* Ask
* Statistical Machine Translation (Stat. MT):



* + Mathematical Description:

**Definition of N-gram models (particular type of LM)**

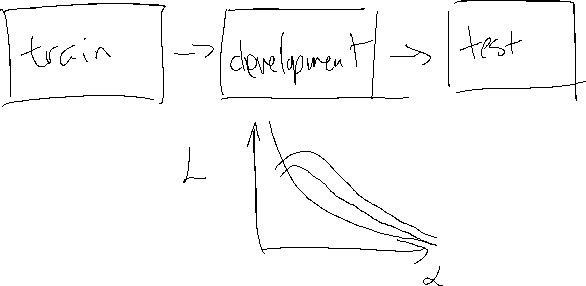
* Given
* Want to find
  + Chain Rule:
  + Markov Assumption:
    - (current word depends on only last words, not all past words)
* Example:
  + Unigram Model:
  + Bigram Model:
* Usually have START and END tokens
* Parameterization ()
  + Bigram Model: parameters
* Steps for Training
  + Collect training set
  + Estimate parameters
  + Test on unseen
* N-gram model; large or small ?
  + Small – may not be enough information
  + Large – hard to train, data is sparse
  + Turns out that for many applications 3 or 4 is good enough
    - Guess the Author was 3-gram model
  + Will see methods in the future for increasing without cons
* How much entropy/ambiguity in English/languages?

**Evaluation**

* Test set
* Look at
* In machine learning, often look at average log likelihood:
* But in NLP, we look at perplexity which satisfies
  + Suppose . Then perplexity must be
  + Lower perplexity does not necessarily mean better performance in application!

**Smoothing**

* Bi-gram model results in sparse data – how to deal with this? Smoothing.
* Zipf’s Law – observation about frequency of words (refer to slides)
* Pseudoscience on Indus Script – many things satisfy Zipf’s law, not just languages
* Smooth sparse data distribution by ‘redistributing the wealth’
  + Add (Laplacian)
    - Assigns a non-zero prior to all words
    - ,
    - How to determine hyperparameter ?



* + Linear Interpolation
    - Combines information from unigrams, bigrams, …, and -grams