Overview of Statistical Language Models

ChengXiang Zhai

Department of Computer Science

University of Illinois, Urbana-Champaign

Outline

- What is a statistical language model (SLM)?
- Brief history of SLM
- Types of SLM
- Applications of SLM

What is a Statistical Language Model (LM)?

- A probability distribution over word sequences
  - p(“*Today is Wednesday*”) ≈ 0.001
  - p(“*Today Wednesday is*”) ≈ 0.0000000000001
  - p(“*The eigenvalue is positive*”) ≈ 0.00001
- Context-dependent!
- Can also be regarded as a probabilistic mechanism for “generating” text, thus also called a “generati

Today is Wednesday

Today Wednesday is

…

The eigenvalue is positive

Definition of a SLM

- Vocabulary set: V={t1, t2, …, tN}, N terms
- Sequence of M terms: s= w1 w2 …wM , wi ∈V
- Probability of sequence s:
  – p(s)=p(w1 w2 … wM)=?
- How do we compute this probability? How do we “generate” a sequence using a probabilistic model?
  – Option 1: Assume each sequence is generated as a “whole unit”
  – Option 2: Assume each sequence is generated by generating one word each time
    - Each word is generated independently
  – Option 3: ??

Brief History of SLMs

- **1950s~1980: Early work,** mostly done by the **IR community**
  - Main applications are to select indexing terms and rank documents
  - Language model-based approaches "lost" to vector space approaches in empirical IR evaluation
  - Limited models developed
- **1980~2000: Major progress** made mostly by the **speech recognition community** and **NLP community**
  - Language model was recognized as an important component in statistical approaches to speech recognition and machine translation
  - Improved language models led to reduced speech recognition errors and improved machine translation results
  - Many models developed!

Brief History of SLMs

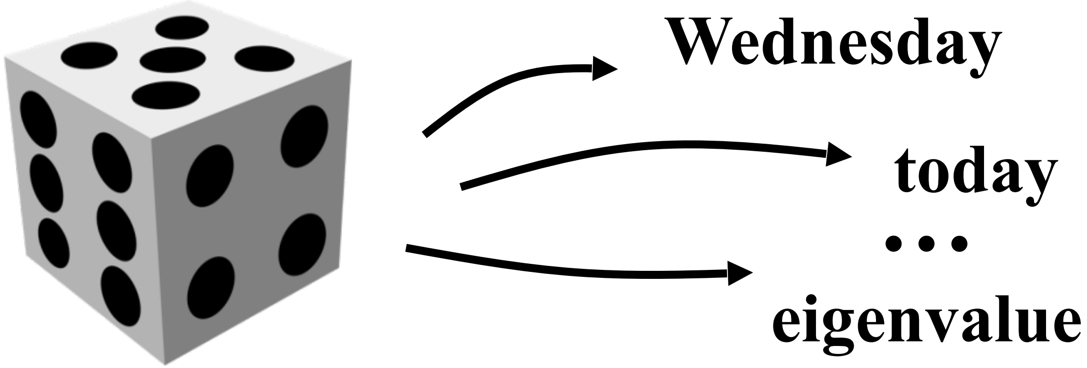
- **1998~2010: Progress made** on using **language models for IR** and for **text analysis/mining**
  - Success of LMs in speech recognition inspired more research in using LMs for IR
  - Language model-based retrieval models are at least as competitive as vector space models with more guidance on parameter optimization
  - Topic language model (PLSA & LDA) proposed and extensively studied
- **2010~ present: Neural language models emerging** and **attracting much attention**
  - Addressing the data sparsity challenge in "traditional" language model
  - Representation learning (word embedding)

Types of SLM

- **“Standard” SLMs** all attempt to formally define p(s) =p(w1….wM)
  – Different ways to refine this definition lead to different types of LMs (= different ways to “generate” text data)
  – Pure statistical vs. Linguistically motivated
  – Many variants come from different ways to capture dependency between words
- **“Non-standard” SLMs** may attempt to define a probability on a transformed form of a text object
  – Only model presence or absence of terms in a text sequence without worrying about different frequencies
  – Model co-occurring word pairs in text
  – …

The Simplest Language Model: Unigram LM

- Generate text by generating each word INDEPENDENTLY
- Thus, $p(w_1\ w_2\ \ldots\ w_n)=p(w_1)p(w_2)\ldots p(w_n)$
- Parameters: $\{p(t_i)\}$ $p(t_1)+\ldots+p(t_N)=1$ (N is voc. size)
- Text = sample drawn according to this **word distribution**



p("today is Wed")

$$= p(\text{"today"})p(\text{"is"})p(\text{"Wed"})$$

$$= 0.0002 \times 0.001 \times 0.000015$$

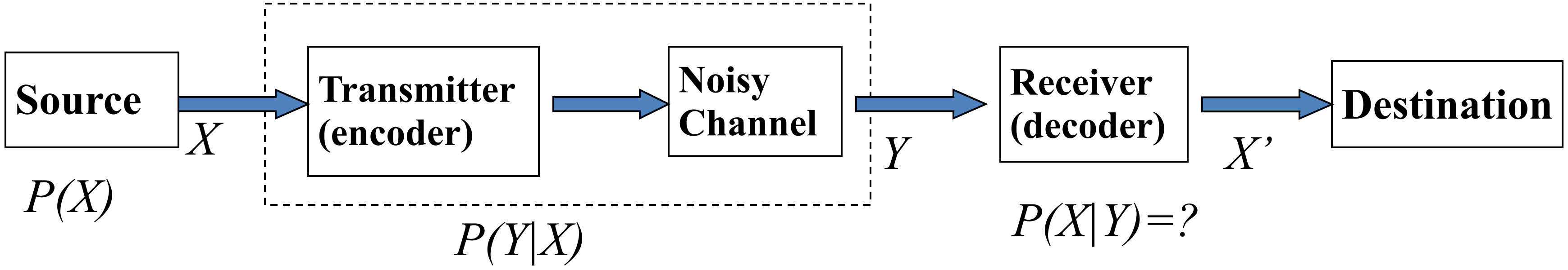
More Sophisticated LMs

- **N-gram language models**
  - In general, $p(w_1\ w_2\ \ldots\ w_n)=p(w_1)p(w_2|w_1)\ldots p(w_n|w_1 \ldots w_{n-1})$
  - n-gram: conditioned only on the past n-1 words
  - E.g., bigram: $p(w_1\ \ldots\ w_n)=p(w_1)p(w_2|w_1)\ p(w_3|w_2)\ \ldots p(w_n|w_{n-1})$
- **Exponential language models** (e.g., Maximum Entropy model)
  - P(w|history) as a function with features defined on "(w, history)"
  - Features are weighted with parameters (fewer parameters!)
- **Structured language models**: generate text based a latent (linguistic) structure (e.g., probabilistic context-free grammar)
- **Neural language models** (e.g., recurrent neural networks, word embedding): model p(w|history) as a neural network

Applications of SLMs

- As a **prior** for **Bayesian inference** when the random variable to infer is text
- As the "**likelihood** part" in **Bayesian inference** when the observed data is text
- As a way to "understand" text data and obtain a more meaningful representation of text for a particular application (**Text Mining**)

Application 1: As Prior in Bayesian Inference:



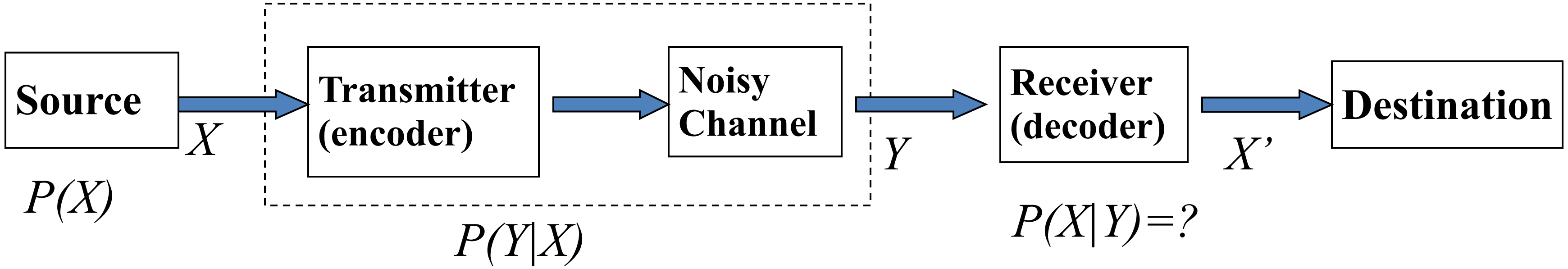
$$\hat{X} = \arg\max_{X} p(X \mid Y) = \arg\max_{X} p(Y \mid X)\, p(X)$$ **(Bayes Rule)**

When X is text, p(X) is a language model

Many Examples:

Speech recognition:	X=Word sequence	Y=Speech signal
Machine translation:	X=English sentence	Y=Chinese sentence
OCR Error Correction:	X=Correct word	Y= Erroneous word
Information Retrieval:	X=Document	Y=Query
Summarization:	X=Summary	Y=Document

Application 2: As Likelihood in Bayesian Inference



$$\hat{X} = \arg\max_{X} \; p(X \mid Y) = \arg\max_{X} \; p(Y \mid X)\, p(X)$$ **(Bayes Rule)**

When Y is text, p(Y|X) is a (conditional) language model

Many Examples:

Text categorization:	X=Topic Category	Y=Text document
Machine translation:	X=English sentence	Y=Chinese sentence
Sentiment tagging:	X=Sentiment label	Y= Text object

Application 3: Language Model for Text Mining

- More interested in the parameters of a language model than the accuracy of the language model itself
  - Parameter values estimated based on a text object or a set of text objects can be directly useful for a task (e.g., topics covered in the text data)
  - Parameter values may serve as a "model-based representation" of text objects to further support downstream applications (e.g., dimension reduction due to representing text by a set of topics rather than a set of words)
- Examples
  - discovery of frequent sequential patterns in text data by fitting an n-gram language model to the text data
  - Part-of-speech tagging & parsing with a SLM

Using Language Models for POS Tagging

Training data (Annotated text)

This	*sentence*	*serves*	*as*	*an*	*example*	*of*	
Det	N	V1	P	Det	N	P	
annotated	*text…*						
V2	N						

"This is a new sentence" → POS Tagger → *This* *is* *a* *new* *sentence* / Det Aux Det Adj N

Consider all possibilities, and pick the one with the highest probability

This	*is*	*a*	*new*	*sentence*
Det	**Det**	**Det**	**Det**	**Det**
… …				
Det	**Aux**	**Det**	**Adj**	**N**
… …				
V2	**V2**	**V2**	**V2**	**V2**

Method 1: Independent assignment
Most common tag

$$p(w_1,\ldots,w_k,t_1,\ldots,t_k) = \begin{cases} p(t_1 \mid w_1)\ldots p(t_k \mid w_k)\, p(w_1)\ldots p(w_k) \\ \prod_{i=1}^{k} p(w_i \mid t_i)\, p(t_i \mid t_{i-1}) \end{cases}$$

Method 2: Partial dependency

w_1="*this*", w_2="*is*", …. t_1=*Det*, t_2=*Det*, …,

Using SLM for Parsing (Probabilistic Context-Free Grammar)

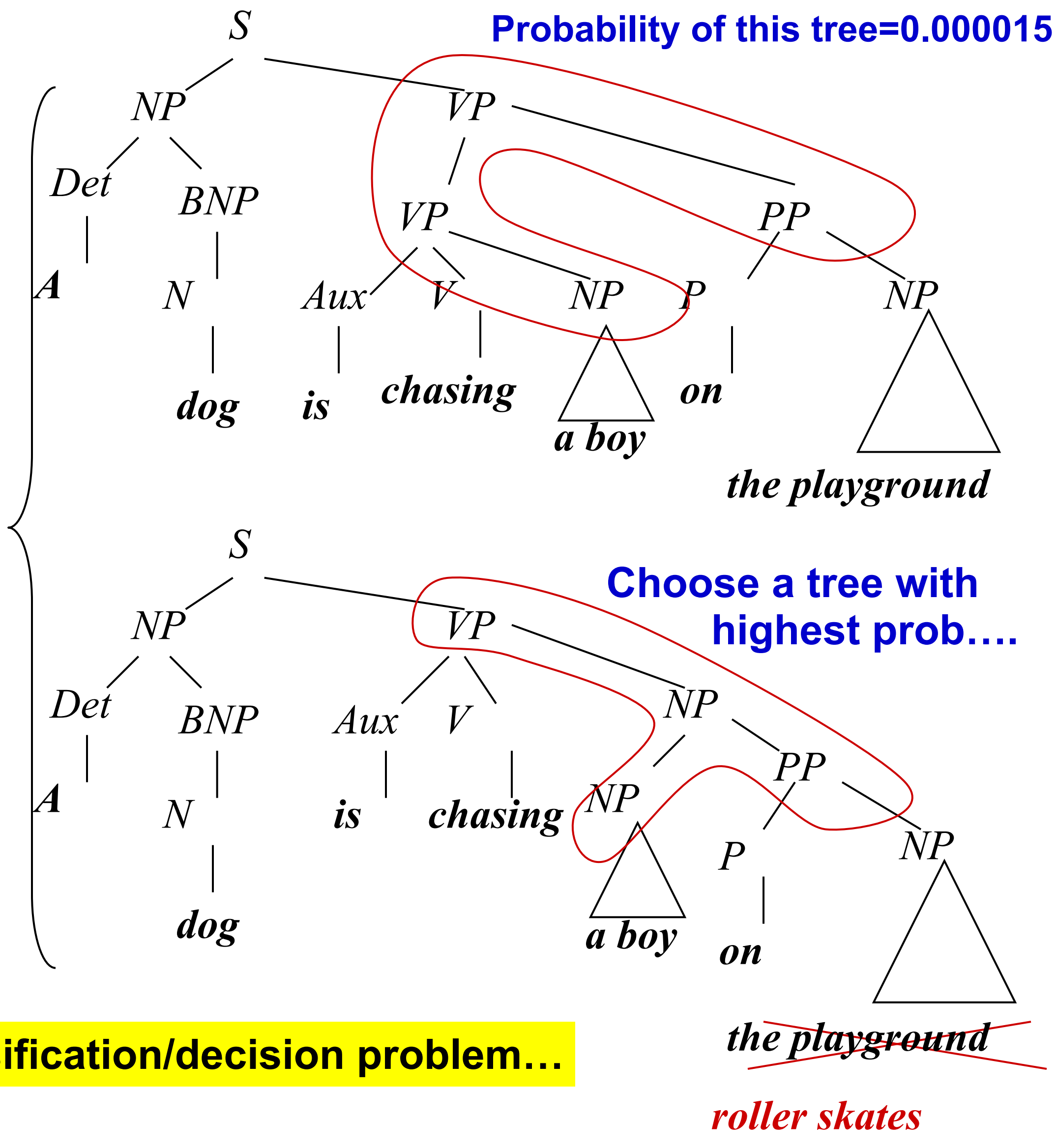
Grammar

$S \rightarrow NP\ VP$	1.0
$NP \rightarrow Det\ BNP$	0.3
$NP \rightarrow BNP$	0.4
$NP \rightarrow NP\ PP$	0.3
$BNP \rightarrow N$	…
$VP \rightarrow V$	
$VP \rightarrow Aux\ V\ NP$	
$VP \rightarrow VP\ PP$	…
$PP \rightarrow P\ NP$	1.0

Lexicon

$V \rightarrow chasing$	0.01
$Aux \rightarrow is$	
$N \rightarrow dog$	0.003
$N \rightarrow boy$	
$N \rightarrow playground$	…
$Det \rightarrow the$	
$Det \rightarrow a$	…
$P \rightarrow on$	

Generate



Can also be treated as a classification/decision problem…

Importance of Unigram Models for Text Retrieval and Analysis

- Words are meaningful units designed by humans and often sufficient for retrieval and analysis tasks
- Difficulty in moving toward more complex models
  – They involve more parameters, so need more data to estimate (A doc is an extremely small sample)
  – They increase the computational complexity significantly, both in time and space
- Capturing word order or structure may not add so much value for “topical inference”, though using more sophisticated models can still be expected to improve performance
- It’s often easy to extend a method using a unigram LM to using an n-gram LM

Evaluation of SLMs

- Direct evaluation criterion: How well does the model fit the data to be modeled?
  - Example measures: Data likelihood, perplexity, cross entropy, Kullback-Leibler divergence (mostly equivalent)
- Indirect evaluation criterion: Does the model help improve the performance of the task?
  - Specific measure is task dependent
  - For retrieval, we look at whether a model helps improve retrieval accuracy, whereas for speech recognition, we look at the impact of language model on recognition errors
  - We hope more "reasonable" LMs would achieve better task performance (e.g., higher retrieval accuracy or lower recognition error rate)

What You Should Know

- What is a statistical language model?
- What is a unigram language model?
- What is an N-gram language model? What assumptions are made in an N-gram language model?
- What are the major types of language models?
- What are the three ways that a language model can be used in an application?