Data Mining and Machine Learning

Language Modelling for Automatic Speech Recognition

Peter Jančovič

Objectives

- Understand role of language model in speech recognition

- Approaches to Language Modelling:

  - Rule-Based Language Models

  - Statistical Language Models
- N-gram Language Models

Speech Recognition: Statistical Methods

- Given an unknown utterance y, want to find the word sequence W such that $P(W \mid y)$ is maximised

By Bayes' Theorem,

$$P(W \mid y) = \frac{p(y \mid W)P(W)}{p(y)}$$

- $P(W)$ - probability that the word sequence W is in application language - **language model probability**

Language Modelling

- **Language Model (Grammar)** used to compute the probability *P(W)* that the sequence of words W 'belongs to' the language

- Constrains recognition problem – fewer possible interpretations

- Basically there are two types of candidate LM:

  - Rule-based (traditional) language model
  - Probabilistic language model

Rule-Based Language Models

- Language models in linguistics and natural language processing typically **rule-based**
- A rule-based language model consists of:
  - A set of **non-terminal units** (e.g. sentence, noun-phrase, verb-phrase, …)
  - A set of **terminal units** (e.g. words)
  - A set of **rules,** defining how non-terminal units can be expanded into sequences of non-terminal and terminal units
- Corresponds to formal notion of grammar – like in school

Rule-Based Language Models

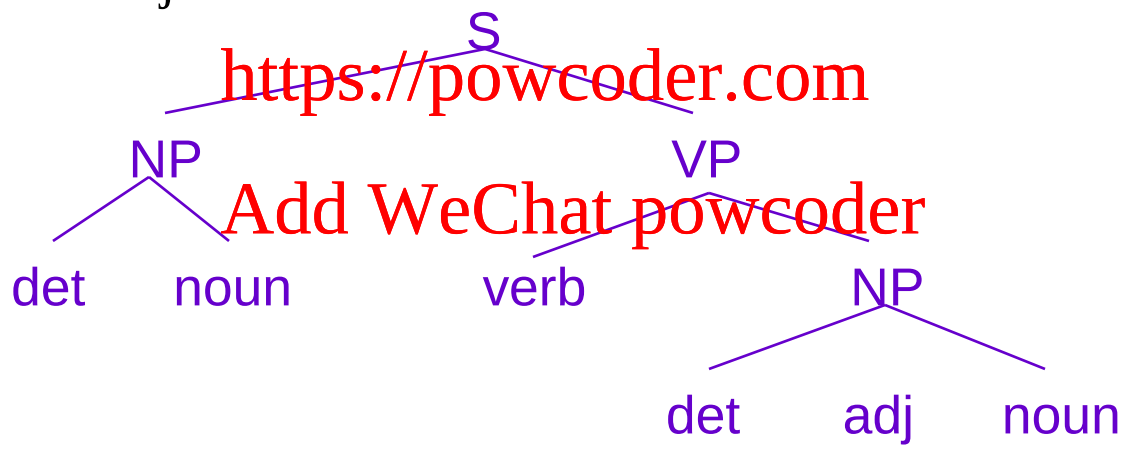
- Let *S* denote the non-terminal root node corresponding to 'sentence'
- A sequence of words is **grammatical** if it can be derived from *S* by a sequence of rules.
- Example: Consider the sentence

  "The cat devoured the tiny mouse"

(From Geoffrey Finch, "How to study linguistics", MacMillan, 1998)

Rule-Based Language Models

- Example rules:
  - S :- NP + VP
  - NP:- det + noun
  - VP:- verb + NP
  - NP:- det + adj + noun
  - det:- “the”
  - noun:- “cat”
  - verb:- “devoured”
  - adj:- “tiny”
  - noun:- “mouse”



The cat devoured the tiny mouse

Rule-Based Language Models

- Disadvantages
  - Normally applied to **written** language
  - A deterministic model of this type may not be able to accommodate variability of **spoken** language
  - Cannot easily handle uncertainty
  - Cannot be derived automatically from example data and is - based on human knowledge

Rule-Based Language Models

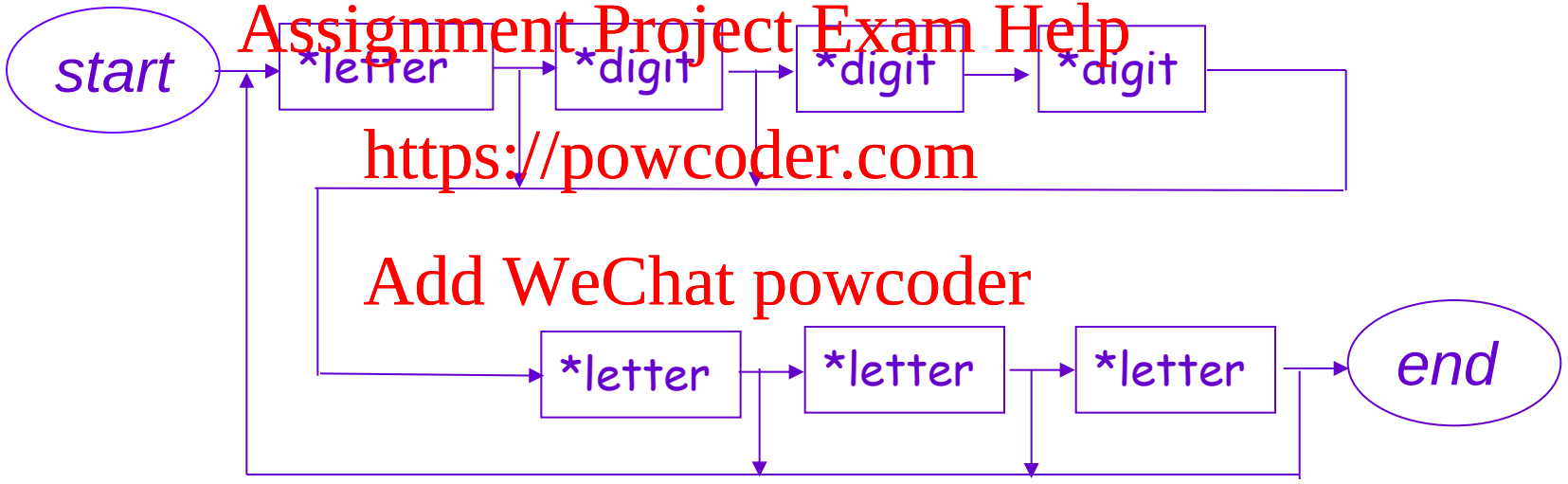
- Advantages
  - Can model complex structure, e.g. non-local dependencies
  - Significant human expertise and knowledge already exists
  - Much effort has already been devoted to the construction of large language models of this type

"She ran, waving enthusiastically, across the bridge"

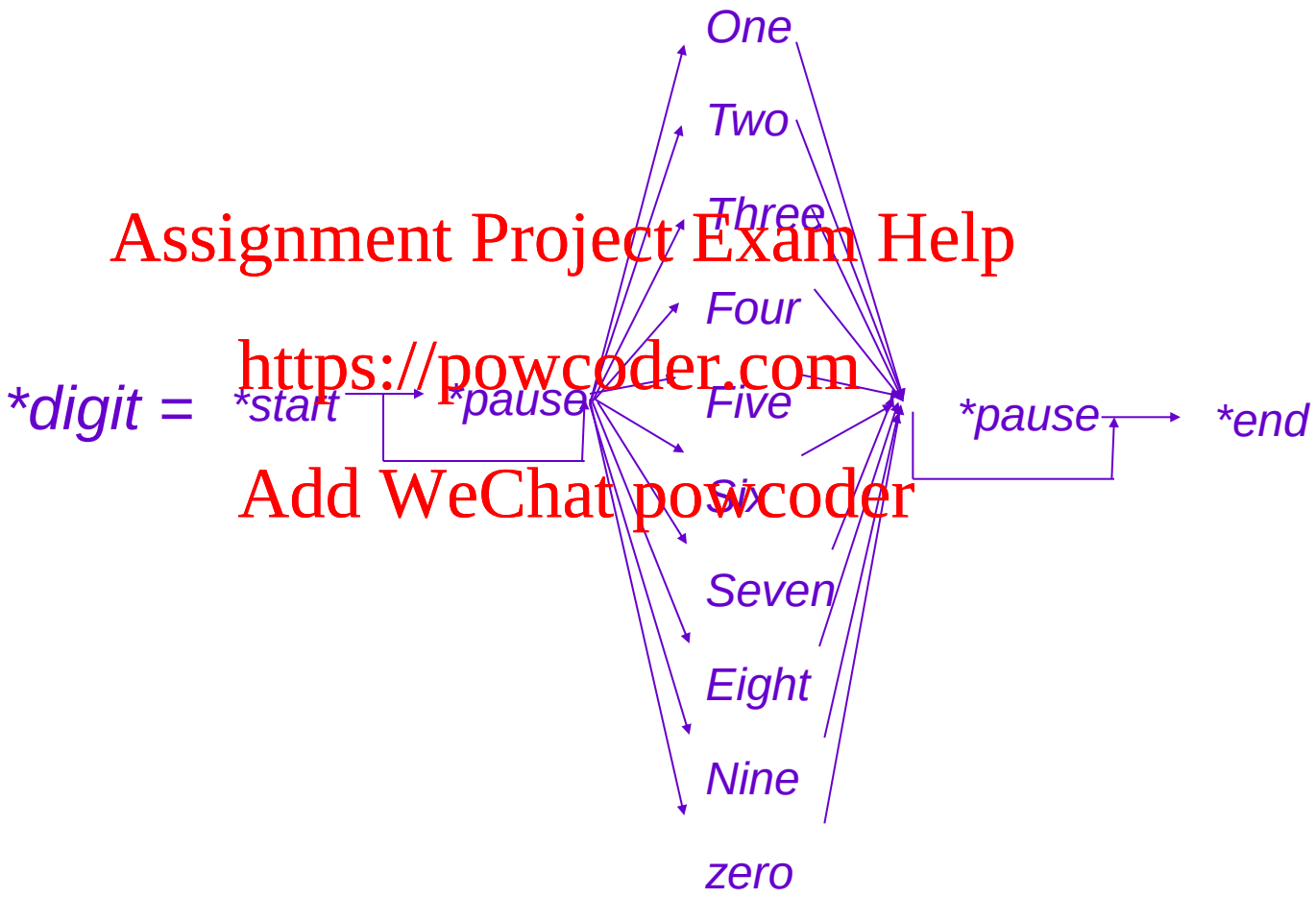
Finite State Language Models

- Describe all possible sentences as routes through a finite state network
- Typically ‘hand-crafted’ using graphical design tools
- Not normally used for vocabulary sizes greater than ~1,000 words

Finite-State Syntax



Expansion of Macros



Statistical Language Models

- With a rule based language model, a sequence of words *W* is either
  - in the language (**grammatical**) or
  - outside the language (**not grammatical**)
- With a statistical language model, a sequence of words *W* is in the language (grammatical) with **probability *P(W)***
- The most common statistical language model is known as the ***N*-gram model**

N-gram Language Models

- Let $W = W_1, W_2, \ldots, W_K$ be a sequence of words
- In general:

$$P(W) = P(W_1)P(W_2|W_1)\ldots P(W_n|W_{n-1},\ldots,W_1)\ldots P(W_K|W_{K-1},\ldots,W_1)$$

- In an *N*-gram language model, we assume:

$$P(W_k|W_{k-1},W_{k-2},\ldots,W_1) = P(W_k|W_{k-1},\ldots W_{k-N+1})$$

i.e. the probability of the k^{th} word in the sequence depends only on identities of the previous *N-1* words

- The most commonly used *N*-gram models are *2*-gram (**bigram**) and *3*-gram (**trigram**) models

Bigram and Trigram Models

- In a **Bigram Language Model**, we assume:

$$P(W_k|W_{k-1},W_{k-2},\ldots,W_1) = P(W_k|W_{k-1})$$

- Similarly, in a **Trigram Language Model**, we assume:

$$P(W_k|W_{k-1},W_{k-2},\ldots,W_1) = P(W_k|W_{k-1},W_{k-2})$$

- These probabilities can be **estimated from data**

Estimation of Bigram Probabilities

- For example, given a training text, an estimate of the bigram probability $P(W_2|W_1)$ is given by:

$$P(W_2|W_1) = N(W_1,W_2)/N(W_1)$$

where:

- $N(W_1,W_2)$ = number of times the word pair W_1,W_2 occurs in the training text

- and $N(W_1)$ = number of times the word W_1 occurs in the training text

Bigram Probabilities - Example

- Consider the training text:

  "*John sat on the old chair. John read the old book. John was interesting. The book was interesting*"

- Suppose this is used to train a bigram grammar.

- *'the'* occurs *3* times in the text, while the bigrams *'the old'* and *'the book'* occur twice and once respectively. Hence

  $P(\text{'old'}|\text{'the'})=2/3$, and $P(\text{'book'}|\text{'the'})=1/3$.

- Similarly, if the symbol # denotes start of sentence, then

  $P(\text{'john'}|\#)=3/4$, and $P(\text{'the'}|\#)=1/4$

Example Continued

- The probability of the sentence *S*

  "John sat on the old chair" is given by:

  $P(S)$

  $= P(john|\#) \cdot P(sat|john) \cdot P(on|sat) \cdot P(the|on) \cdot P(old|the)$
  $\cdot P(chair|old) \cdot P(\$|chair)$

  $= 3/4 \cdot 1/3 \cdot 1 \cdot 1 \cdot 2/3 \cdot 1/2 \cdot 1 = 1/12$

- Similarly

  $P(\text{"The old chair"}) = 1/12$

  $P(\text{"John read the old chair"}) = 1/12$

- But $P(\text{"John read the interesting book"}) = 0$

Bigram & Trigram Estimation

- Most practical systems use a trigram language model
- In reality, there is never enough text to estimate trigram probabilities in this simple way
- E.g. experiments with trigram language models for a 1,000 word vocabulary application
  - using 1.5 million words for training, and 300,000 words to test the models
  - 23% of the trigrams in the **test** corpus were absent from the **training** corpus
- Hence much more sophisticated training procedures are needed

Estimation of *N*-gram statistics

- In general, there will not be enough data to estimate *N*-gram statistics reliably.
- Possible solutions:
  - Robust estimation methods for statistics
  - Deleted interpolation
  - ‘Back-off’

Deleted interpolation

- 'Interpolate' trigram probability from estimated trigram, bigram and unigram probabilities:

$$\hat{P}(w_3 \mid w_2 w_1) \approx \lambda_1 P(w_3 \mid w_2 w_1) + \lambda_2 P(w_3 \mid w_2) + \lambda_3 P(w_3)$$

- Estimate λ_1, λ_2, λ_3 through recognition experiments

‘Backoff’

- Decide how many examples T are needed for robust estimation.
- Then:

$$\hat{P}(w_3 \mid w_2 w_1) = \begin{cases} P(w_3 \mid w_2 w_1) & \text{if } |w_1 w_2 w_3| \geq T \\ P(w_3 \mid w_2) & \text{if } |w_1 w_2 w_3| < T \text{ and } |w_1 w_2| \geq T \\ P(w_3) & \text{otherwise} \end{cases}$$

N-gram Language Models - Summary

- Advantages
  - Can be trained automatically from data
  - Probabilistic model
  - Consistent with acoustic model
  - Mathematically sound algorithms

N-gram Language Models - Summary

- Disadvantages
  - Large amounts of training data needed
  - Difficult to incorporate human knowledge
  - Cannot model long term dependency:
    "She walked, hands in pockets, quickly across the bridge"

Summary

- Role of language modelling in speech recognition
- Rule based language models
- Finite state language models
- *N*-gram language models
- Difficulty of estimating *N*-gram statistics