FLST: Statistical Language Models

Dietrich Klakow
Dietrich.Klakow@LSV.Uni-Saarland.de

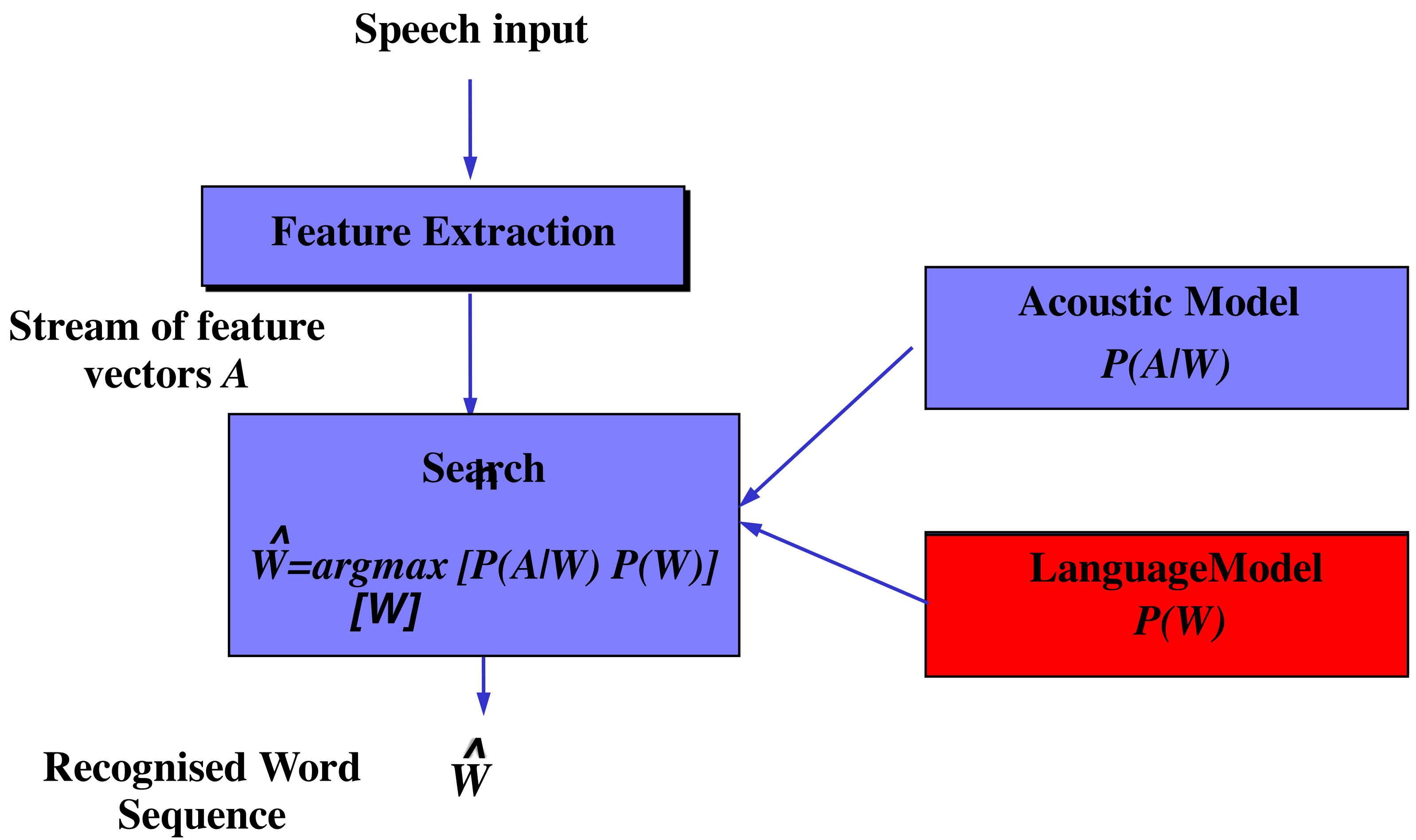
Goal

Some of the lecture material is covered in other lectures.

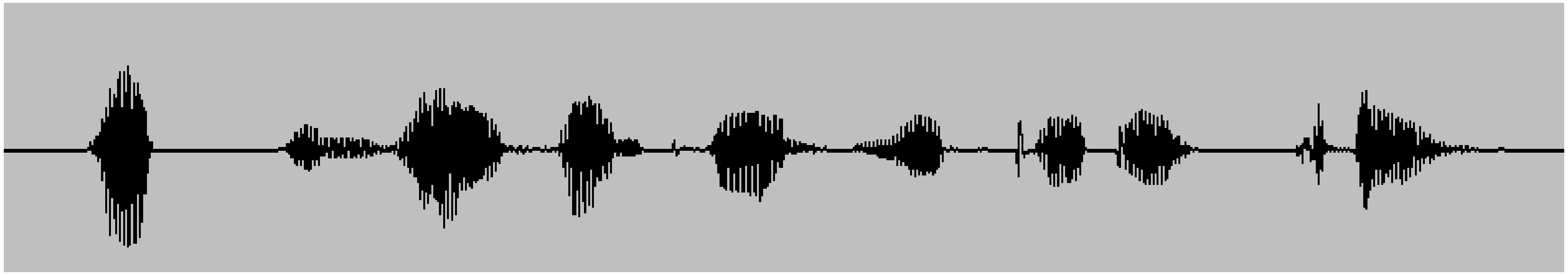
Goal: understand all the details such that you have running implementation in the end.

Using Language Models

How Speech Recognition works

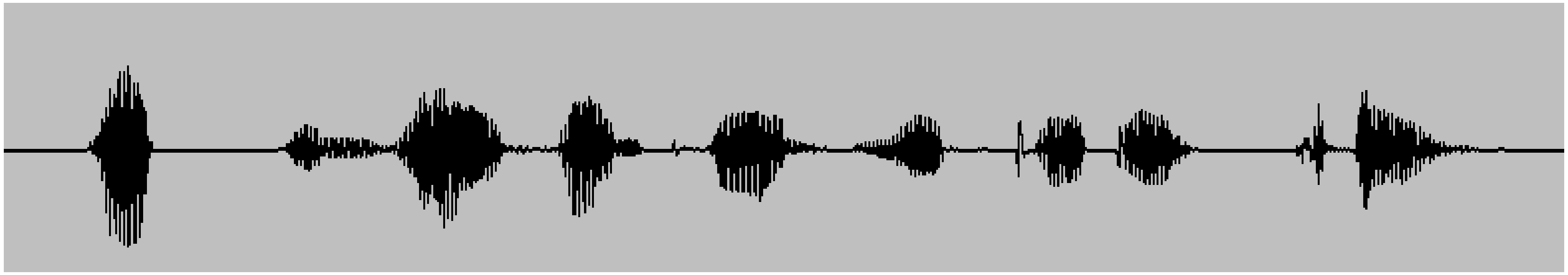


Guess the next word



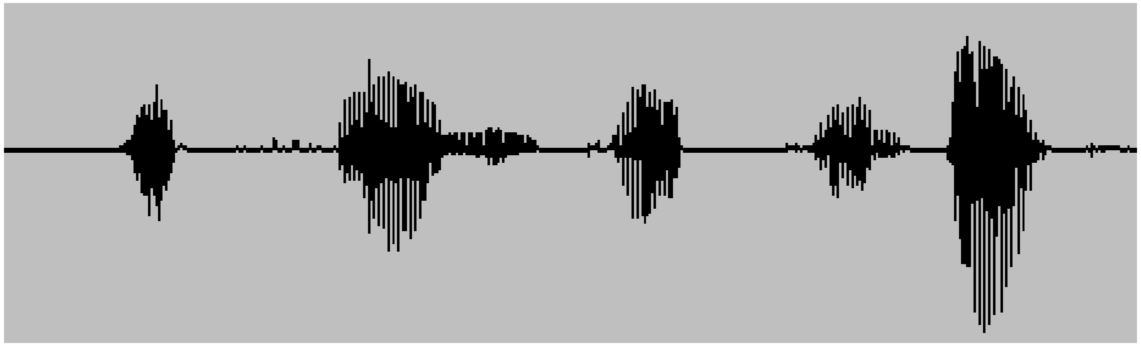
What‘s in your hometown newspaper ???

Guess the next word



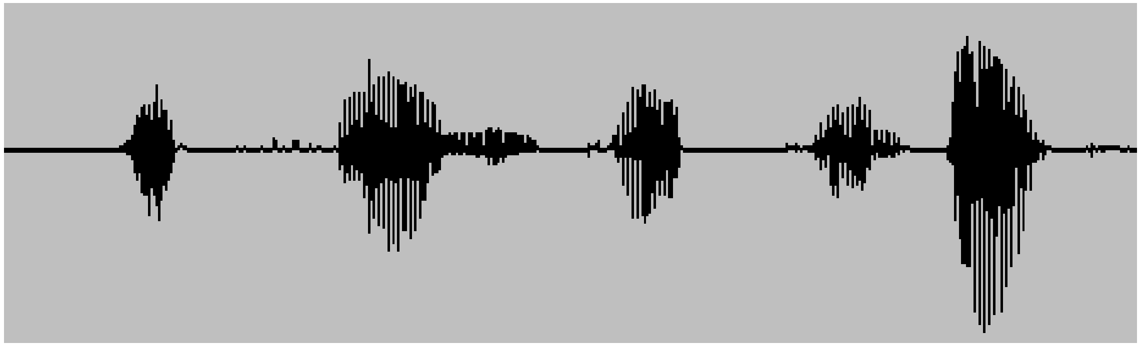
What's in your hometown newspaper today

Guess the next word



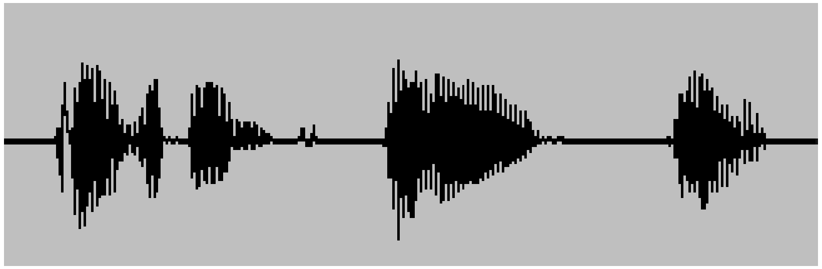
It‘s raining cats and ???

Guess the next word



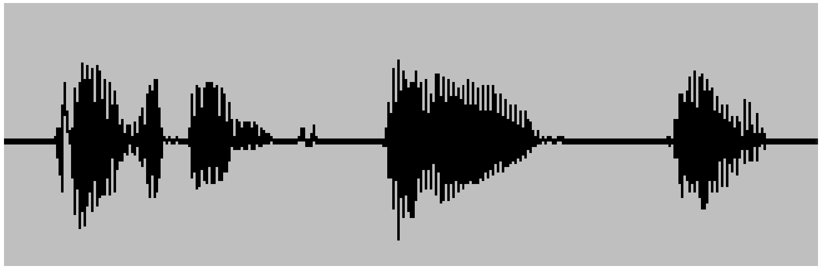
It‘s raining cats and dogs

Guess the next word



President Bill ???

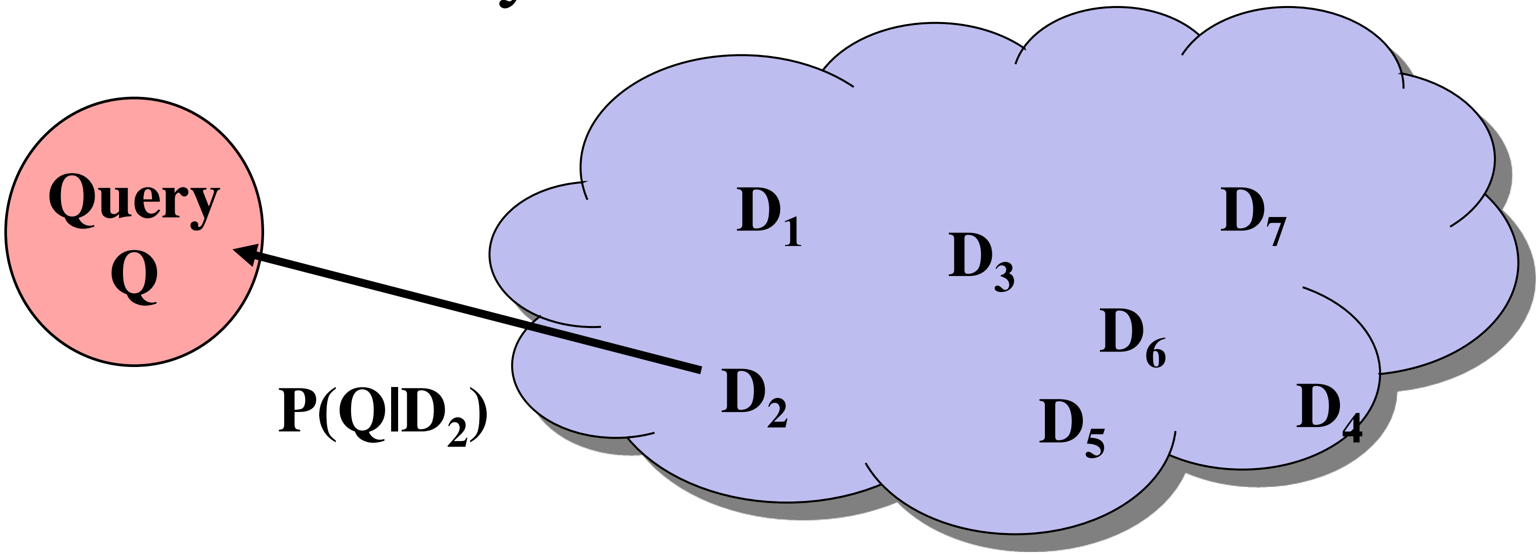
Guess the next word



President Bill Gates

Information Retrieval

- Language model introduced to information retrieval in 1998 by Ponte&Croft



Ranking according to $P(Q|D_i)$

Measuring the Quality of Language Models

Definition Perplexity

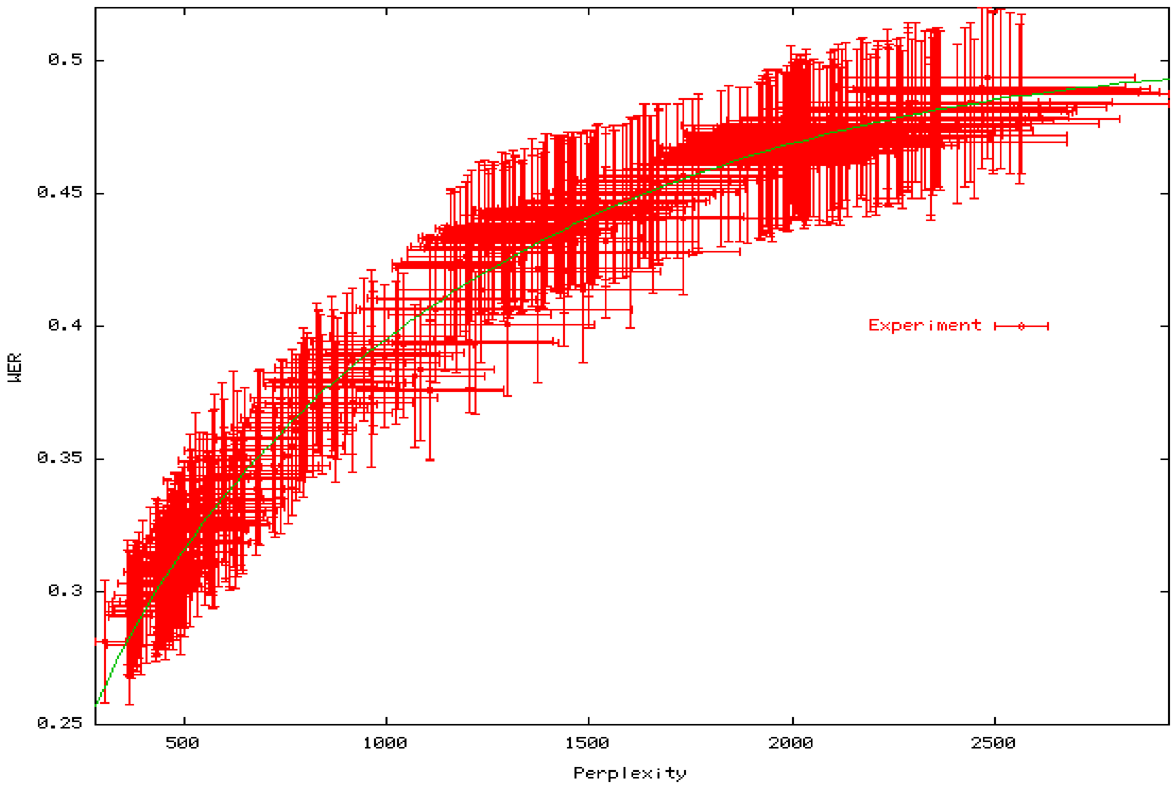
$$PP = P(w_1 \ldots w_N)^{-1/N}$$
$$= \exp\left(-\frac{1}{N} \sum_{w,h} N(w,h) \log \left(P(w \mid h) \right) \right)$$

P(w|h):	language model
N(w,h):	frequency of sequence w,h in some test corpus
N:	size of test corpus

Calculate perplexity of uniform distribution

(white board)

Perplexity and Word Error Rate



Perplexity and error rate are correlate within error bars

Estimating the Parameters of a Language Model

Goal

- Minimize perplexity on training data

$$\boxed{PP = \exp\left(-\frac{1}{N_{Train}} \sum_{w,h} N_{Train}(w,h) \log\left(P(w \mid h)\right) \right)}$$

Define Likelihood

L=-log (PP)

$$\boxed{L = \frac{1}{N_{Train}} \sum_{w,h} N_{Train}(w,h) \log \left(P(w \mid h) \right)}$$

Minimizing perplexity
$\mapsto$
maximizing likelihood

How to take normalization constraint into account?

Calculating the maximum likelihood estimate

(white board)

Maximum Likelihood Estimate

$$\boxed{P(w \mid h) = \frac{N_{Train}(w,h)}{N_{Train}(h)}}$$

What s the problem?

Backing-off and Smoothing

Absolute Discounting

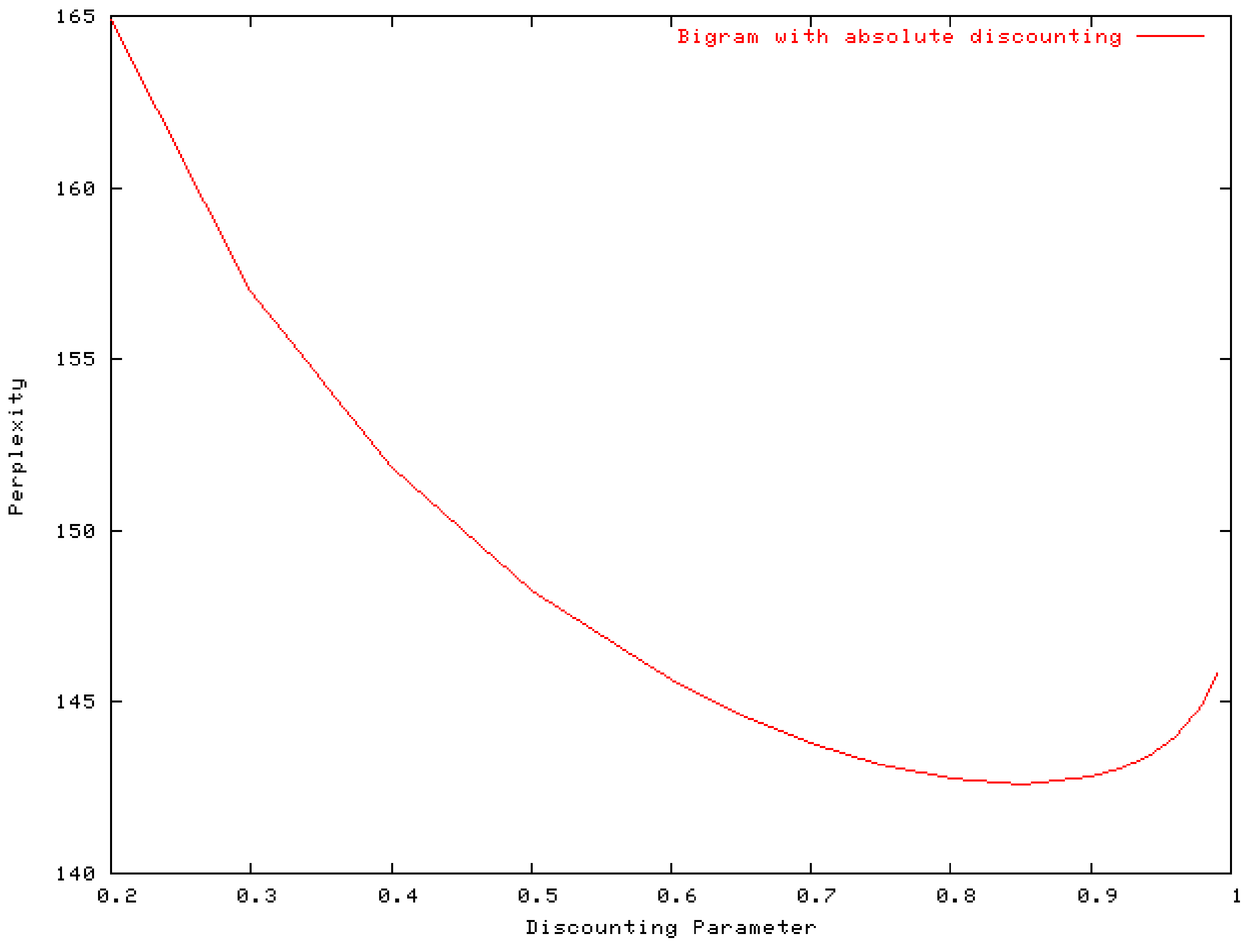
See white board

Backing-Off Language Model

$$P(w\,|\,h)=\begin{cases}\dfrac{N(w,h)-d}{N(h)}+\alpha(h)P\,(w\,|\,\hat{h}) & \text{für } N(w,h)>0\\ \alpha(h)\;P\,(w\,|\,\hat{h})\end{cases}$$

Backing-off weight α(h)
Backing-off distribution P(w|h^) (normalized! Recursive:
again backing off structure with shortened history h^)
Discouting parameter d

Influence of Discounting Parameter



Possible further Improvements

Linear Smoothing

$$P(w_0 \mid w_{-1}) = \lambda_1 \frac{N_{Train}(w_{-1} w_0)}{N_{Train}(w_{-1})} + \lambda_2 \frac{N_{Train}(w_0)}{N_{Train}} + (1 - \lambda_1 - \lambda_2) \frac{1}{V}$$

V: size of vocabulary

Marginal Backing Offf (Kneser Ney Smoothing)

Dedicated backing-off distributions

Usually about 10% to 20% reduction in perplexity

Class Language Models

Automatically group words into classes

Map all words in the language model to classes

Dramatic reduction in number of parameters to estimate

Usually used in linear with word language model

Summary

How to build a state-of-the art plain vanilla language model:

- Trigram
- Absolute discounting
- Marginal backing-off (Kneser-Ney smoothing)
- Linear interpolation with class model