Lecture: HMM Adaptation for ASR

by Tor André Myrvoll

Outline

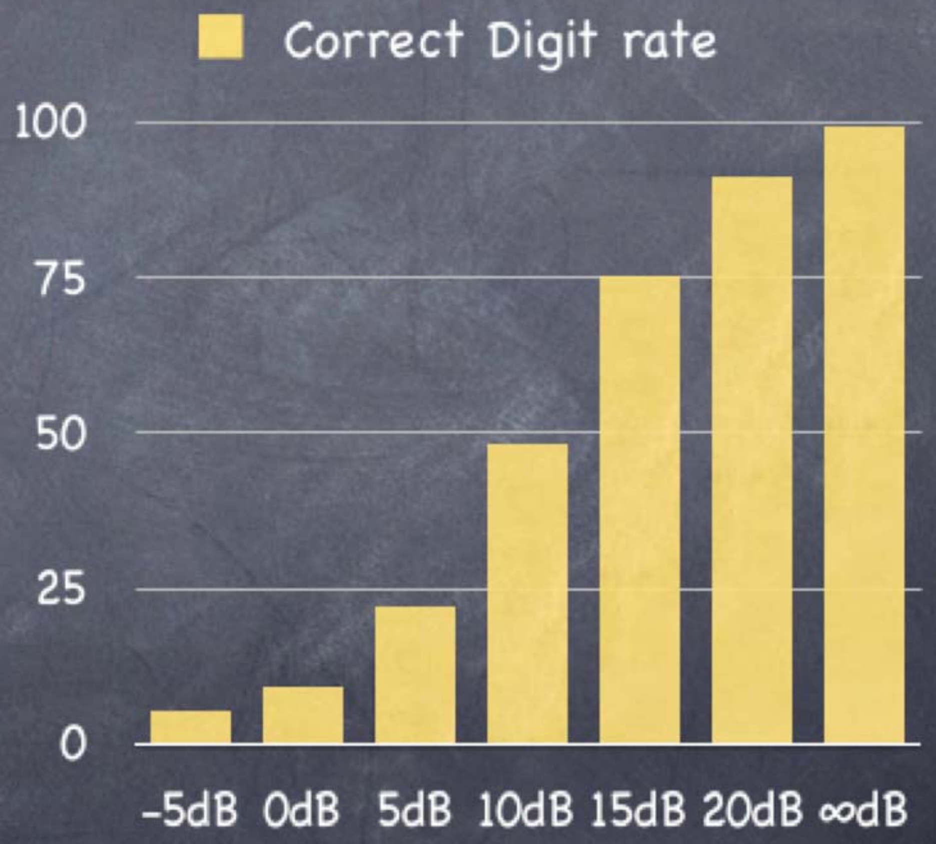
- Motivation: Why use adaptation?
- Adaptation paradigms.
- MAP and MLLR adaptation.
- On-line adaptation.

Motivation

- Mismatches between training and test conditions can severely degrade the ASR performance.
- Specialized models, e.g. speaker dependent models, perform better that more general models.
- Adapted models can be used for bootstrapping model training in another domain.

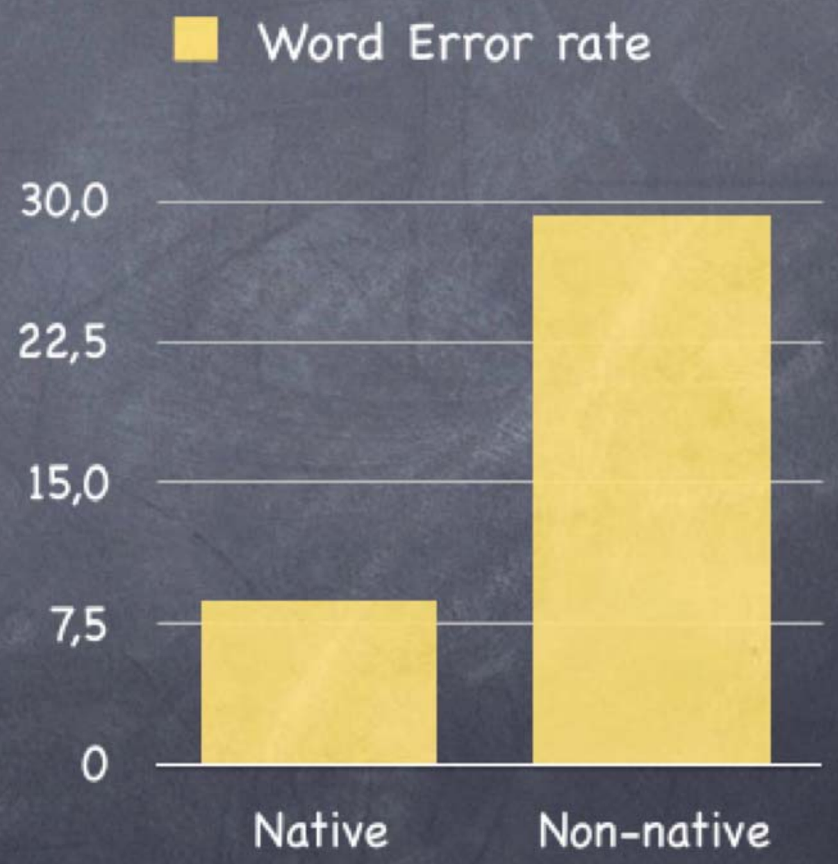
Mismatches: Additive Noise

- Additive noise is an important problem.
- The example to the left shows the degradation of a digit string recognizer under various signal-to-noise ratios.



Mismatches: Speaker variation

- Different speakers have different:
    - Speaking apparatus.
    - Speaking rates.
    - Accents/dialects.
- Extreme case: Non-native speakers.



Adaptation vs. Robustness.

- The term robustness is mostly used in conjunction with additive noise or channel variations.
- This mismatch can modeled and compensated for using a variety of techniques.
- The term adaptation is often used when no mathematical description of the mismatch is available.

Adaptation = Learning

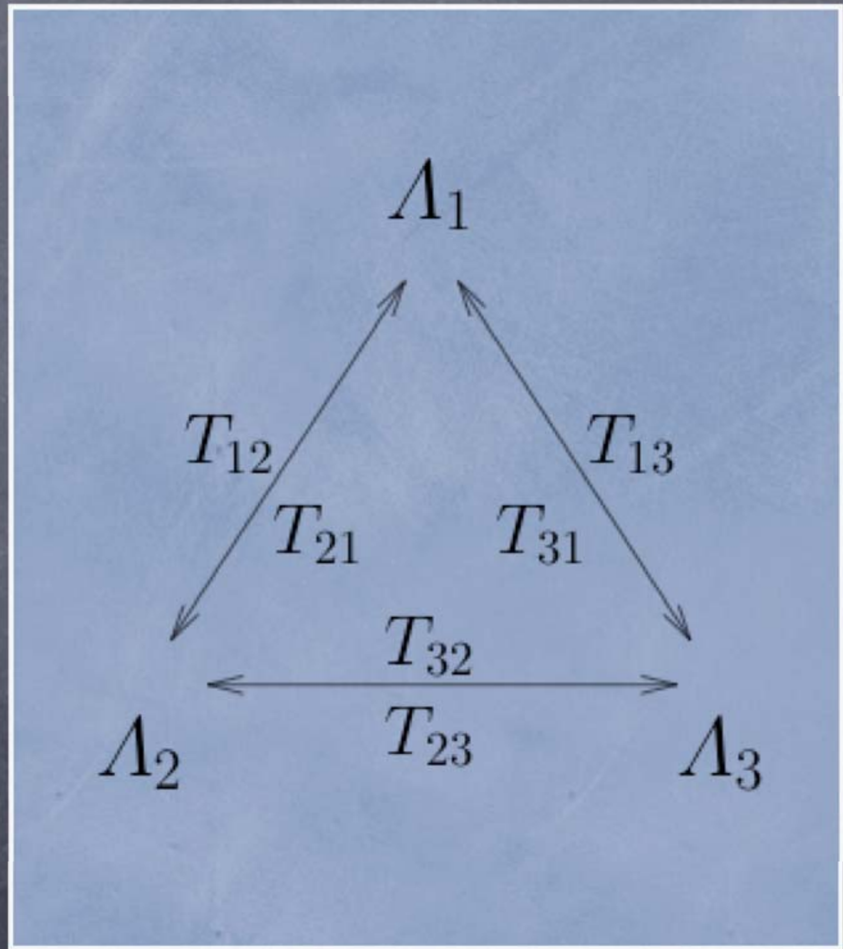
- Learning is what we do when we build an HMM from scratch using training data.
- When no mathematical model of the mismatch is available, we have to learn the mismatch.
- Such learning should not be done from scratch, as the cost would be prohibitive, but using a small amount of representative data.

Adaptation paradigms

- To make adaptation practical, some strong assumptions have to be made.
  - Transformation based adaptation assumes that models are related through simple mappings of HMM parameters.
  - Bayesian approaches assumes that HMM parameters are distributed according to some simple probability density function.

Transformation based adaptation

- The models are assumed to be simple transformations of each other.
- Given a model and a transformation, we can find a new model.
- Goal: Learn the transformations.

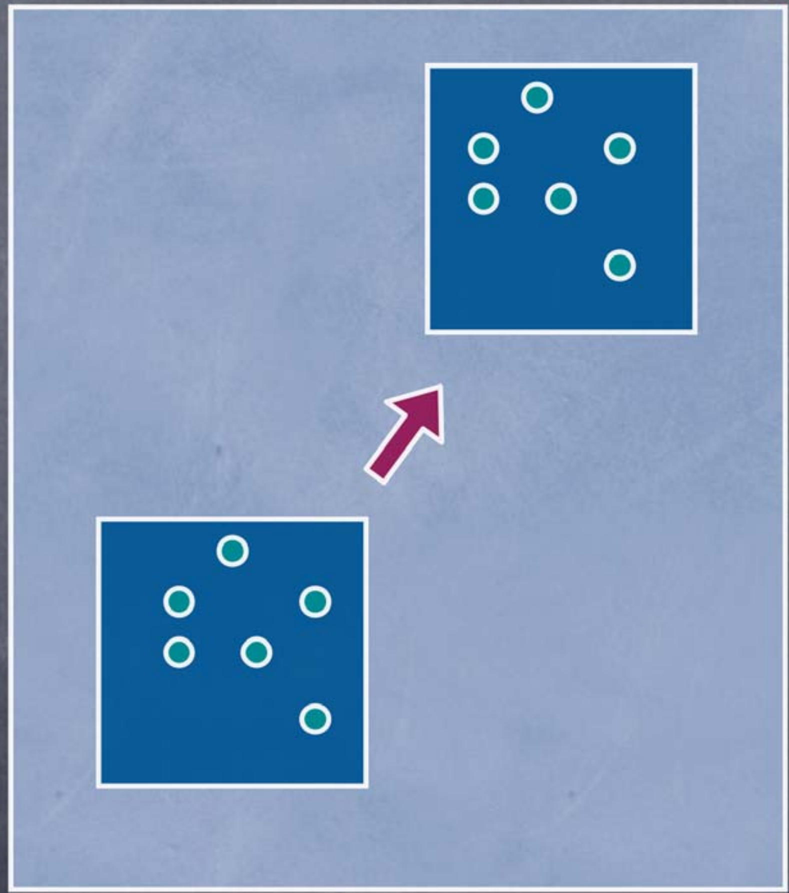


Transformation examples: Bias

- One of the simplest transformations that has been suggested is the mean vector bias.

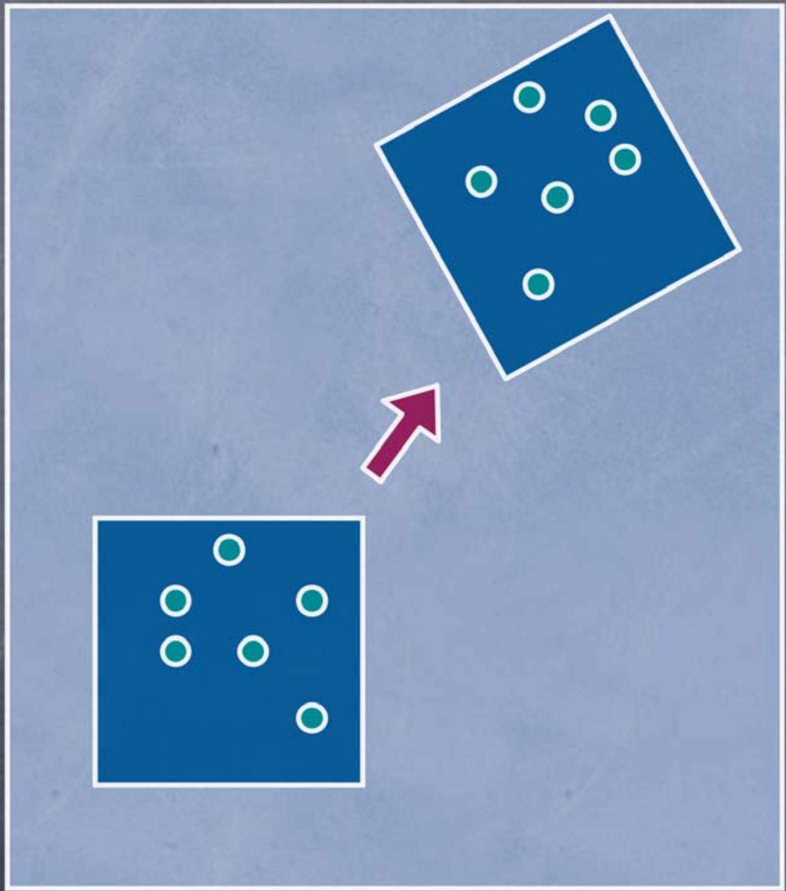
- For every state s and mixture m, the new mean vector is given as

    - $\mu(s,m) \rightarrow \mu(s,m)+b$



Transformation examples: MLLR

- MLLR - Maximum Likelihood Linear Regression.
- For every state s and mixture n, the new mean vector is given as
    - $\mu(s,m) \rightarrow A\mu(s,m)+b$



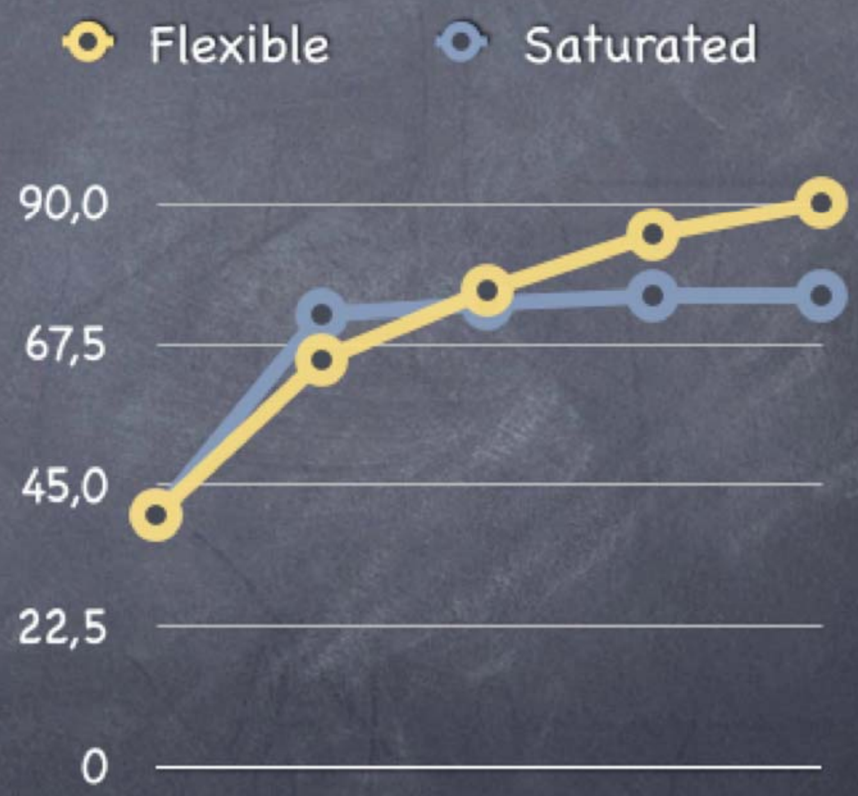
Transformation estimation

- Transformations are usually estimated in a maximum likelihood sense.
- Given some adaptation data O, we want the transformation that maximizes the likelihood of the data:

$$\hat{T} = \mathrm{argmax}_T \, p(O|\Lambda, T)$$

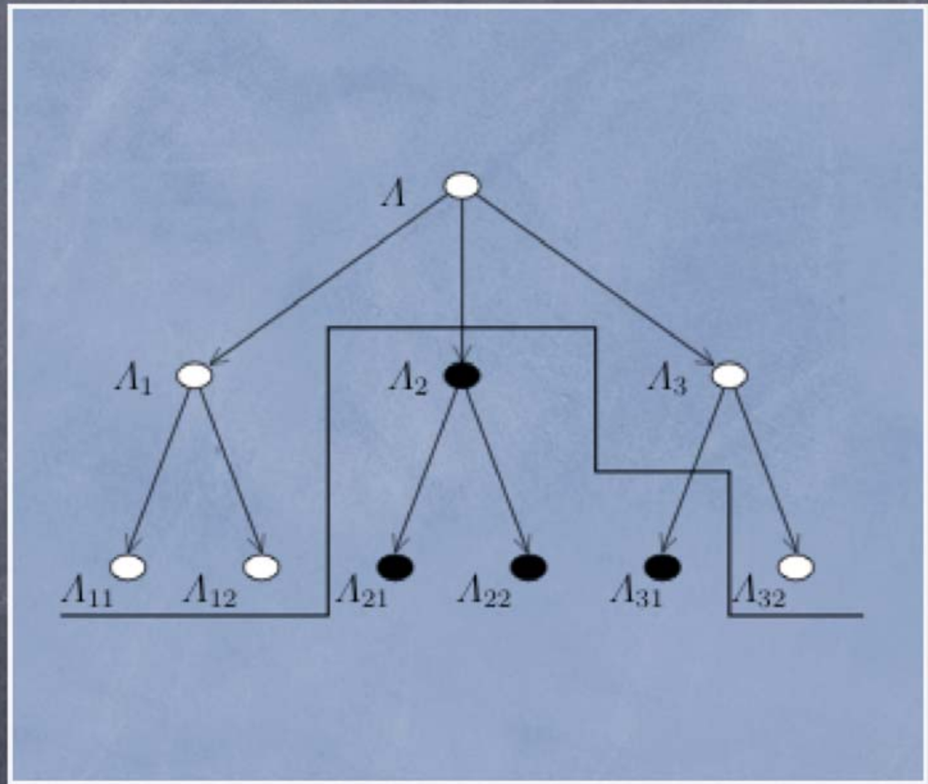
Flexible transformations

- Any adaptation method should make use of all the available data.
- Using a single transformation leads to premature performance saturation



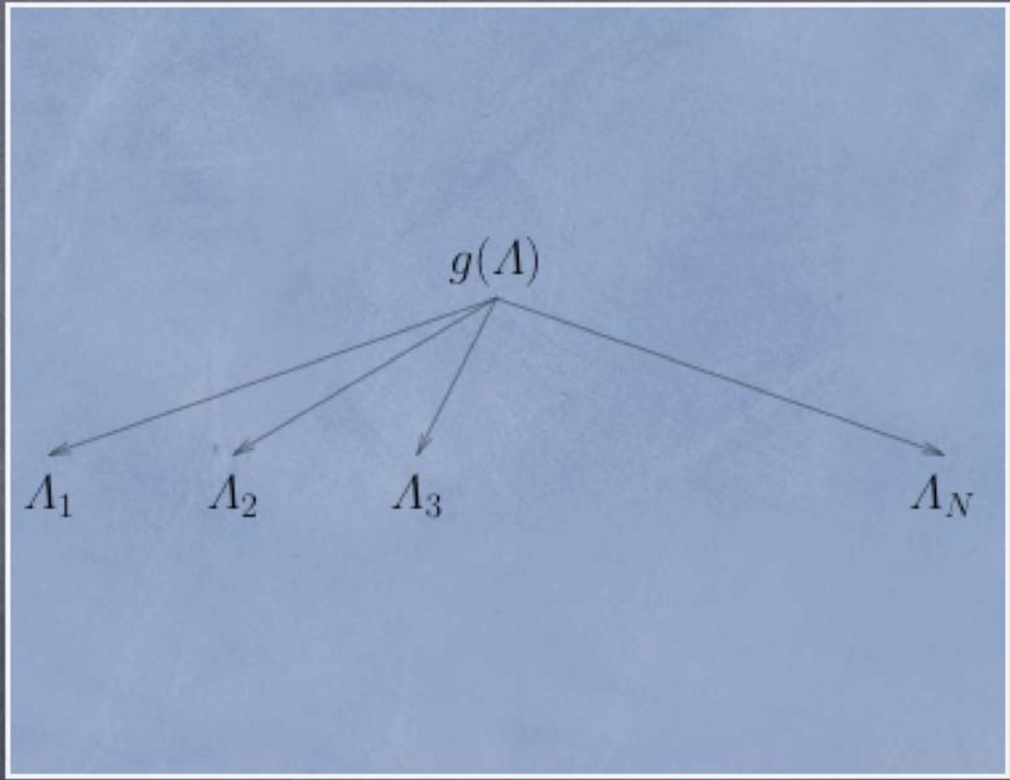
Flexible Transformations

- One solution is to use more than one transformation.
- The number of transformations is chosen according to the amount of data.



Bayesian adaptation

- Every HMM parameter set Λ is assumed to be a drawn according to a pdf g(Λ).
- According to Bayesian statistics, g(Λ) is a prior of the model Λ.



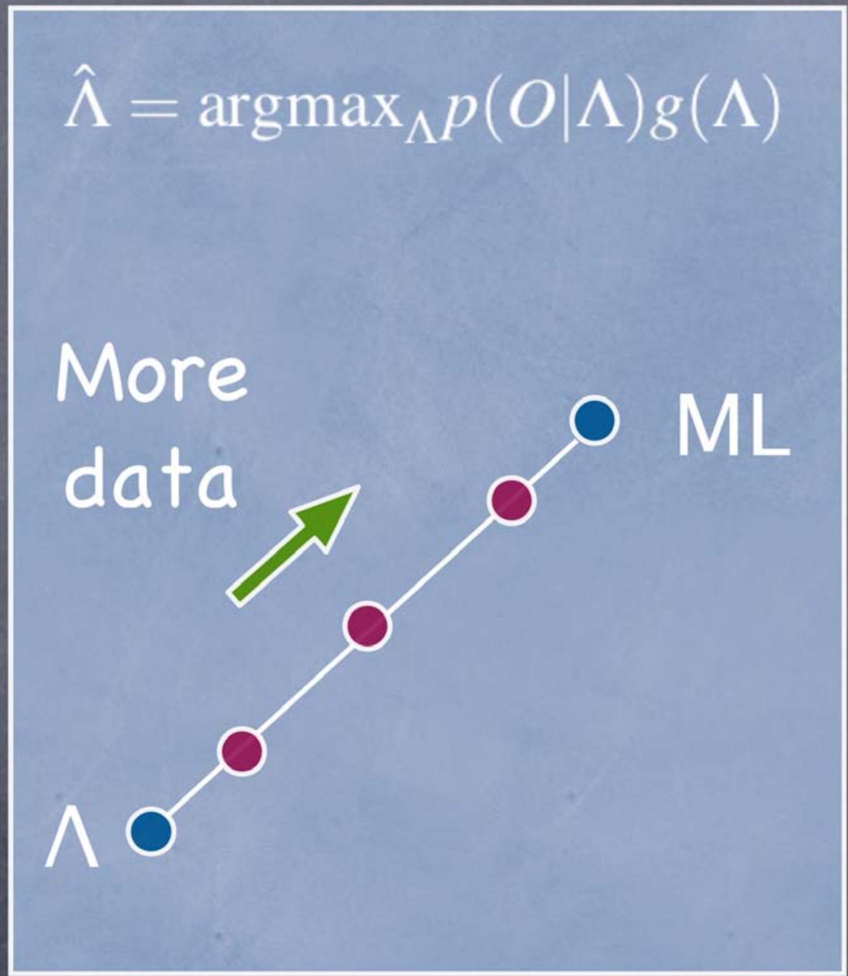
Bayesian adaptation

- When a prior is available we can maximize the probability of the model given the adaptation data, rather than the other way around.
- This is called an maximum a posteriori (MAP) estimate:

$$\hat{\Lambda} = \operatorname{argmax}_{\Lambda} p(O|\Lambda) g(\Lambda)$$

Bayesian adaptation

- Given a relevant prior, Bayesian adaptation is both robust and flexible.
- However, it is a problem to find good priors.
- Modeling correlations is especially problematic.

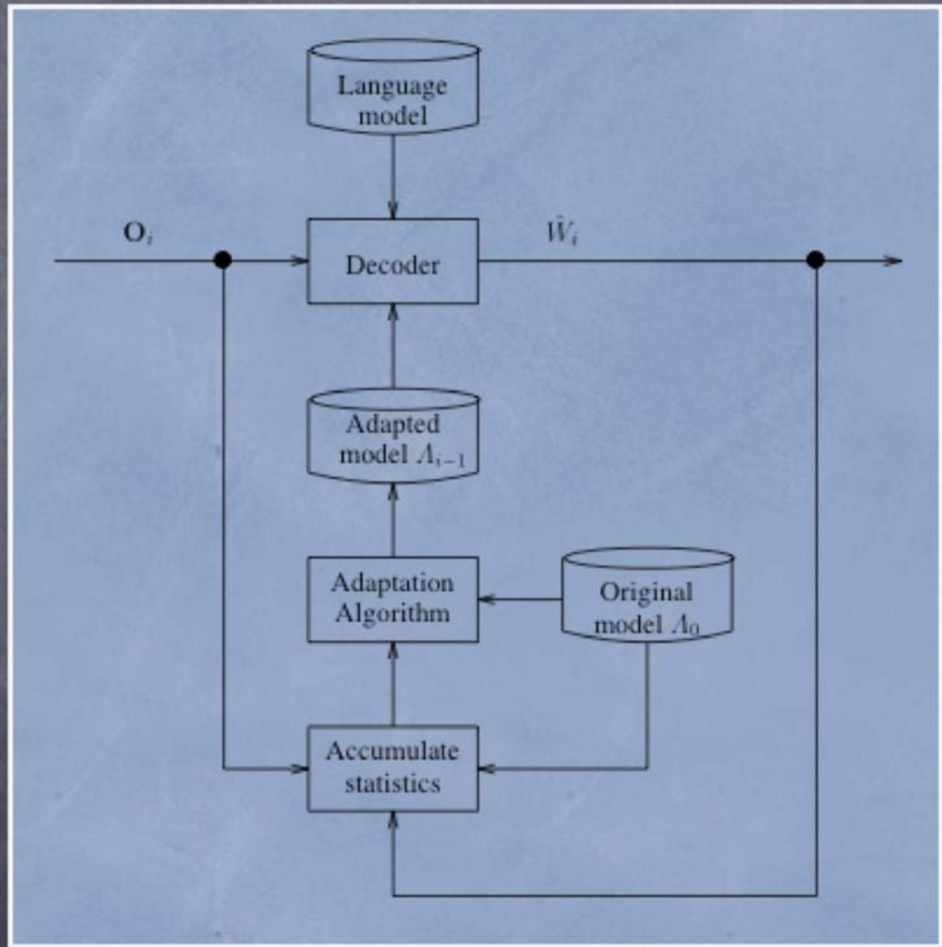


Online adaptation

- Online adaptation is used when we need the system to evolve with changing conditions.
- Several new problems occur:
    - The adaptation data is untranscribed.
    - There is no way to guarantee that the adaptation data is good -> Risk of further degradation.
    - System complexity increases.

Online adaptation

- Simple approach: Accumulate adaptation data and create new model regularly.
- Need to keep two models in memory at a time.
- Won't track continuous changes.



Online adaptation

- Quasi-Bayesian approach. The maximum of the prior is used as a model.
- The prior is updated regularly using the posterior pdf w.r.t. the adaptation data.
- One model in memory.