Smoothing probabilistic automata: an error-correcting approach

P. Dupont[1] and J.-C. Amengual[2]

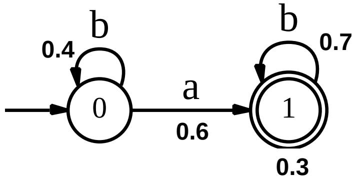
[1] Jean Monnet University - Saint-Etienne - France
[2] Universidad Jaume I - Castellón - Spain

Probabilistic inference: motivations

- Generalization from positive examples is often made under the control of negative examples.
  What can we do if negative examples are not available?
- What can we do in the presence of noise?

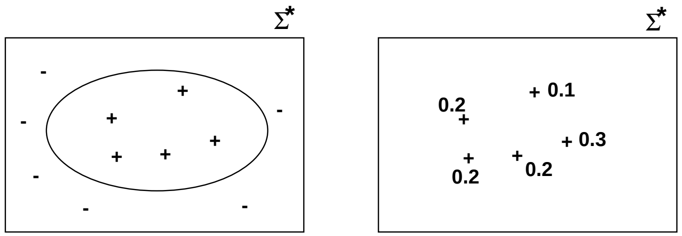
Induce a probabilistic automaton is one way to handle both problems.

Probabilistic Deterministic Finite Automaton (PDFA)



$$P(abb) = \underset{a}{0.6} \times \underset{b}{0.7} \times \underset{b}{0.7} \times \underset{END}{0.3}$$

PDFA inference: two complementary viewpoints



Probabilistic automaton inference algorithms

Carrasco and Oncina 94 ALERGIA

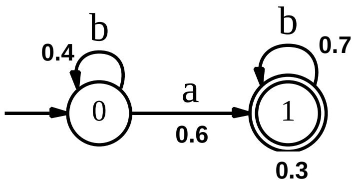
Stolcke 94 *Bayesian* learning

Ron et al. 95 learning of *acyclic* automata

Carrasco and Oncina 99 RLIPS

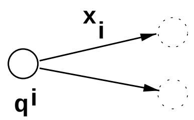
Thollard et al.00 MDI

The PDFA smoothing problem

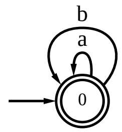


$$P(baa) = \underset{b}{0.4} \times \underset{a}{0.6} \times \underset{a}{0.0} = 0$$

Smoothing by linear interpolation



$\gamma(q^i, x_i)$: probability of generating x_i from state q_i



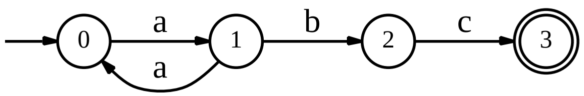
$P_1(x_i)$ probability of generating x_i from a unigram model

Smoothing by linear interpolation:

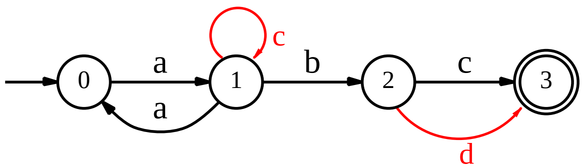
$$P(x_i|q^i) = \beta\gamma(q^i, x_i) + (1-\beta)P_1(x_i)$$

Error-correcting smoothing

P(a c b d) = ?

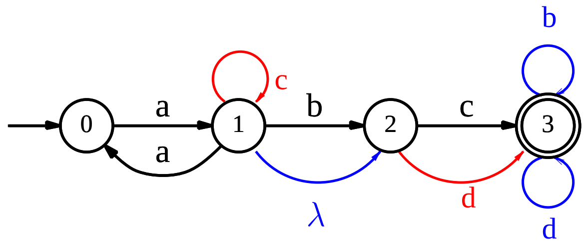


P(a c b d)



Error-correcting smoothing(2)

$P(\mathrm{a}\ \lambda\ \mathrm{c}\ \mathrm{b}\ \mathrm{d})$

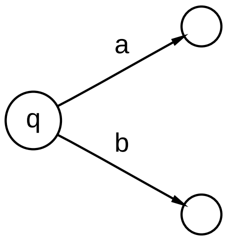


non-error (original) transitions

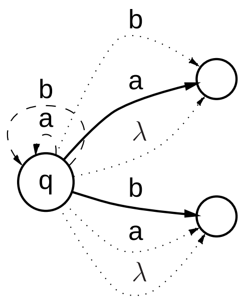
error transitions

error transitions

Error-correcting model



Original automaton



Extended automaton

$P(\lambda \to a \mid q)$ probability of inserting symbol a in state q

$P(a \to b \mid q, q')$ probability of substituting a by b while going from q to q'

$P(a \to \lambda \mid q, q')$ probability of deleting a while going from q to q'

State-independent error model

$$P(\lambda \to a \mid q) \triangleq P(\lambda \to a)$$

$$P(a \to b \mid q, q') \triangleq P(a \to b)$$

$$P(a \to \lambda \mid q, q') \triangleq P(a \to \lambda)$$

- reduction of the number of parameters
- symbol *dependent* but state/transition *independent* error model
- possible combination of several error models independently of the automaton structure

Extended automaton distribution

Original transition probability of symbol a from state q (counts computed on the training sample) :

$\gamma(q, a) = \frac{C_q(a)}{C_q}$

Error counts computed on a *independent* sample :

$C(\lambda, a)$ number of insertions of a

$C(a, b)$ number of substitutions of a by b

$C(a, \lambda)$ number of deletions of a

Insertion probability: $P(\lambda \to a) \propto C(\lambda, a)$

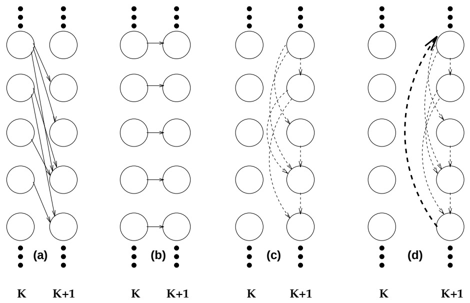
Substitution and non-error transition probability:
$P(a \to b) \cdot \gamma(q, a) \propto C(a, b) \cdot \gamma(q, a)$

Deletion probability: $P(a \to \lambda) \cdot \gamma(q, a) \propto C(a, \lambda) \cdot \gamma(q, a)$

Greedy estimation of an error model

1. Infer a PDFA from a training sample (e.g. using ALERGIA)
2. Set initial error counts and define the extended automaton
3. For each string of an *independent* sample, compute the most likely path through the extended automaton
4. Compute the insertion, substitution and deletion counts of each symbol along these paths
5. Update the error model according to the new counts
6. Goto 3 unless some stopping criterion is met

Computation of the most likely path



(a) substitution and non-error transitions
(b) insertion
(c) deletion in an acyclic PDFA
(d) deletion in a cyclic PDFA

⇒ extension of the Viterbi algorithm (see [Amengual and Vidal98])

Experiments: the ATIS task

Air Travel Information System

The corpus consists of speakers of American English making spoken requests such as

"Uh, I'd like to go from, uh, Pittsburgh to Boston next Tuesday, no wait, Wednesday".

Lexicon (Σ): 1294 words

Learning set: 13044 sentences, 130773 tokens

Validation set: 974 sentences, 10636 tokens

Test set: 1001 sentences, 11703 tokens

Quality measure: the perplexity

$P(x_i^j|q^i)$: probability of generating x_i^j, the i-th symbol of the j-th string from state q_i

$$LL = \left(-\frac{1}{\|S\|} \sum_{j=1}^{|S|} \sum_{i=1}^{|x|} \log P(x_i^j|q^i) \right)$$

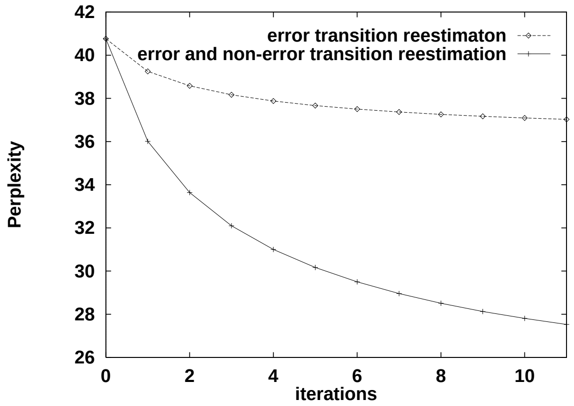
$$PP = 2^{LL}$$

Perplexity PP measures the predictive power of a model

$PP = 1$: a perfectly predictive model

$PP = |\Sigma|$: uniform random guessing from an alphabet Σ

Error-model reestimation on validation set



Estimation by cross-validation

- Concatenate training and validation set in a single data set.
- Construct N (typically 10) different partitions of this data set.
- For each partition, infer a PDFA on the first part (typically 90 % of the data set) and estimate an error model on the second part (typically the remaining 10 %).
- Merge all error models by summing up the error counts obtained on each partition.

Test set perplexity

Type I error model: error transition reestimation

Type II error model: error and non-error transition reestimation

CV: Cross-validation

Model	Perplexity
Unigram smoothing ($\beta = 0.5$)	71
Type I error model	40
Type II error model	41
CV + Type I error model	37
CV + Type II error model	37

Perspectives

- Repeat the experiments with another inference algorithm (e.g. MDI)
- Refine the error model by adding a weight between error versus non-error transitions for each state
- Apply the same smoothing technique to non-deterministic automata
- Combine error-correcting smoothing with symbol clustering