Novel algorithms for minimum risk phrase chunking

Martin Jansche

11/11, 11:30

1 Examples of chunks

Fat chunks: base NPs

$\boxed{\textit{De liberale minister}}$ *van* $\boxed{\textit{Justitie}}$ $\boxed{\textit{Marc Verwilghen}}$ *is* $\boxed{\textit{geen kandidaat}}$ *op* $\boxed{\textit{de lokale VLD-lijst}}$ *bij* $\boxed{\textit{de komende gemeenteraadsverkiezingen}}$ *in* $\boxed{\textit{Dendermonde}}$.

$\{(1,3),(5,5),(6,7),(9,10),(12,14),(16,18),(20,20)\}$

Lean chunks: named entities

De liberale minister van Justitie Marc Verwilghen is geen kandidaat op de lokale VLD-lijst bij de komende gemeenteraadsverkiezingen in Dendermonde.

$G = \{(5,5),(6,7),(14,14),(20,20)\}$

2 Evaluation

- Number of true positives:

$$|G \cap H|$$

- Precision P:

$$\frac{|G \cap H|}{|H|}$$

- Recall R:

$$\frac{|G \cap H|}{|G|}$$

The standard definitions of precision and recall ignore the limiting cases where $G = \emptyset$ or $H = \emptyset$.

Gloss over this detail by redefining the following indeterminate form:

$$\frac{0}{0} \overset{\mathrm{def}}{=} 1$$

3 Evaluation example

Gold standard: *De liberale minister van* $\boxed{\textit{Justitie}}$ $\boxed{\textit{Marc Verwilghen}}$ *is geen kandidaat op de lokale* $\boxed{\textit{VLD-lijst}}$ *bij de komende g23n in* $\boxed{\textit{Dendermonde}}$.

Hypothesis: *De liberale minister van* $\boxed{\textit{Justitie Marc Verwilghen}}$ *is geen kandidaat op de lokale* $\boxed{\textit{VLD-lijst}}$ *bij de komende gemeenteraadsverkiezingen in* $\boxed{\textit{Dendermonde}}$.

$$P = \frac{2}{3} \qquad R = \frac{2}{4}$$

4 The overall tasks

1. Given unlabeled text, find all the chunks (maximize recall) and only the chunks (maximize precision).
2. Given labeled text, infer the best possible chunker (perhaps to an approximation).

Do not lose sight of these overall tasks!

5 The task-specific loss function

Effectiveness measure E (van Rijsbergen, 1974):

$$E = 1 - \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

For $\alpha = 0.5$ this amounts to

$$E = \frac{|G \cup H - G \cap H|}{|G| + |H|}$$

The goal is to minimize this loss, in a sense to be made precise.

6 The central reduction

1. Treat the processing task as a sequence labeling task.

2. Treat the learning task as a sequence learning task.

Do this by annotating each token with a label drawn from the set

$$\Gamma = \{\mathrm{I}, \mathrm{O}, \mathrm{B}\}$$

using the IOB2 annotation scheme (Ratnaparkhi, 1998; Tjong Kim Sang and Veenstra, 1999).

De liberale minister van Justitie Marc Verwilghen is geen kandidaat op de lokale VLD-lijst bij de komende gemeenteraadsverkiezingen in Dendermonde.

De liberale minister van Justitie Marc Verwilghen is geen kandidaat op de lokale VLD-lijst bij de komende gemeenteraadsverkiezingen in Dendermonde.

becomes

De/O liberale/O minister/O van/O Justitie/B Marc/B Verwilghen/I is/O geen/O kandidaat/O op/O de/O lokale/O VLD-lijst/B bij/O de/O komende/O gemeenteraadsverkiezingen/O in/O Dendermonde/B ./O

Formally, this is a mapping from a word sequence w plus a set of non-overlapping chunks to w plus a label sequence x.

The set of valid IOB2 label sequences is the following star-free language:

$$\{\mathsf{O}, \mathsf{B}\}\Gamma^* - \Gamma^*\{\mathsf{OI}\}\Gamma^*$$

Formally, this is a mapping from a word sequence w plus a set of non-overlapping chunks to w plus a label sequence x.

The set of valid IOB2 label sequences is the following star-free language:

$$\{\mathsf{O}, \mathsf{B}\}\Gamma^* - \Gamma^*\{\mathsf{OI}\}\Gamma^*$$

The number of label sequences of length n is

$$\Theta((\phi + 1)^n)$$

where $\phi = (1 + \sqrt{5})/2$ is the Golden Ratio.

7 The sequence labeling tasks

A supervised instance consists of a sequence of words $w = (w_1, \ldots, w_n)$ together with a corresponding sequence of labels $x = (x_1, \ldots, x_n)$.

Note that $|w| = |x|$. This is what makes the present task relatively straightforward, compared to e.g. speech recognition.

The processing task consists of finding a label sequence x given a word sequence w. The learning task consists of inferring a suitable model on the basis of training samples of the form (w, x).

We want to model the probability of a label sequence x given a word sequence w. We can do so indirectly via a generative model, e.g. an HMM

$$\Pr(w, x) = \prod_{i=1}^{n} \Pr(w_i \mid x_i) \times \Pr(x_i \mid x_{i-1})$$

We want to model the probability of a label sequence x given a word sequence w. We can do so indirectly via a generative model, e.g. an HMM

$$\Pr(w, x) = \prod_{i=1}^{n} \Pr(w_i \mid x_i) \times \Pr(x_i \mid x_{i-1})$$

or directly, using a conditional model

$$\Pr(x \mid w) = \prod_{i=1}^{n} \Pr(x_i \mid x_{i-1}, w)$$

(Ratnaparkhi, 1998; Bengio, 1999; McCallum et al., 2000). The precise choice of model does not matter at this point.

8 Loss and utility

Given label sequences x and y, define the following concepts:

- $tp(x, y)$ is the number of matching chunks (true positives) shared by x and y;
- $m(x)$ is the number of chunks in x;
- $P(x \mid y) = tp(x, y)/m(x)$ is precision;
- $R(x \mid y) = tp(x, y)/m(y)$ is recall;

Van Rijsbergen's loss function E with $\alpha \in [0; 1]$

$$E_\alpha(x \mid y) = 1 - \left(\alpha \, \frac{1}{P(x \mid y)} + (1 - \alpha) \, \frac{1}{R(x \mid y)} \right)^{-1}$$

Van Rijsbergen's loss function E with $\alpha \in [0;1]$

$$E_\alpha(x \mid y) = 1 - \left(\alpha \, \frac{1}{P(x \mid y)} + (1-\alpha) \, \frac{1}{R(x \mid y)}\right)^{-1}$$

is replaced by $F_\beta = 1 - E_{1/(\beta+1)}$ with $\beta \in [0;\infty]$:

Van Rijsbergen's loss function E with $\alpha \in [0;1]$

$$E_\alpha(x \mid y) = 1 - \left(\alpha \, \frac{1}{P(x \mid y)} + (1-\alpha) \, \frac{1}{R(x \mid y)}\right)^{-1}$$

is replaced by $F_\beta = 1 - E_{1/(\beta+1)}$ with $\beta \in [0;\infty]$:

$$F_\beta(x \mid y) = \left(\frac{1}{\beta+1} \, \frac{m(x)}{tp(x,y)} + \frac{\beta}{\beta+1} \, \frac{m(y)}{tp(x,y)}\right)^{-1}$$

Van Rijsbergen's loss function E with $\alpha \in [0;1]$

$$E_\alpha(x \mid y) = 1 - \left(\alpha \, \frac{1}{P(x \mid y)} + (1-\alpha) \, \frac{1}{R(x \mid y)}\right)^{-1}$$

is replaced by $F_\beta = 1 - E_{1/(\beta+1)}$ with $\beta \in [0;\infty]$:

$$\begin{aligned} F_\beta(x \mid y) &= \left(\frac{1}{\beta+1} \, \frac{m(x)}{tp(x,y)} + \frac{\beta}{\beta+1} \, \frac{m(y)}{tp(x,y)}\right)^{-1} \\ &= \frac{(\beta+1) \, tp(x,y)}{m(x) + \beta \, m(y)} \end{aligned}$$

Van Rijsbergen's loss function E with $\alpha \in [0;1]$

$$E_\alpha(x \mid y) = 1 - \left(\alpha \, \frac{1}{P(x \mid y)} + (1-\alpha) \, \frac{1}{R(x \mid y)}\right)^{-1}$$

is replaced by $F_\beta = 1 - E_{1/(\beta+1)}$ with $\beta \in [0;\infty]$:

$$\begin{aligned} F_\beta(x \mid y) &= \left(\frac{1}{\beta+1} \, \frac{m(x)}{tp(x,y)} + \frac{\beta}{\beta+1} \, \frac{m(y)}{tp(x,y)}\right)^{-1} \\ &= \frac{(\beta+1)\, tp(x,y)}{m(x) + \beta\, m(y)} \end{aligned}$$

Instead of minimizing the loss function E, one can maximize the utility function F.

9 Expected utility

Because the state of the world, y, is not known exactly and only specified by $\mathrm{Pr}(y)$, we take (conditional) expectations:

- $\mathsf{E}[tp(x,\cdot)] = \sum_y tp(x,y)\,\mathrm{Pr}(y \mid w)$ is the expected number of true positives;

- $\mathsf{E}[P(x \mid \cdot)] = 1/m(x) \sum_y tp(x,y)\,\mathrm{Pr}(y \mid w)$ is the expected precision;

- $\mathsf{E}[R(x \mid \cdot)] = \sum_y tp(x,y)/m(y)\,\mathrm{Pr}(y \mid w)$ is the expected recall;

Most importantly,

$$\begin{aligned}\mathcal{U}(x \mid w;\theta) &= \mathsf{E}[F_\beta(x \mid \cdot)] \\ &= (\beta+1)\sum_y \frac{tp(x,y)}{m(x)+\beta\, m(y)} \Pr(y \mid w;\theta)\end{aligned}$$

is the expected utility of a label sequence x given a word sequence w.

This is the objective (as a function of x) we want to maximize when searching for the best hypothesis, according to the Bayes Decision Rule (see e.g. Duda et al., 2000).

10 The processing task

- Assume a fully specified probability model with known parameter vector θ has been fixed.

- Given a word sequence w, we want to find chunks. (That's the overall task.)

- Finding chunks has been reduced to assigning a label sequence.

- Assign ("decode") the label sequence

$$\hat{x} = \operatorname*{argmax}_{x} \mathcal{U}(x \mid w; \theta)$$

Often in the NLP literature, the decoding step is replaced by

$$\tilde{x} = \operatorname*{argmax}_{x} \Pr(x \mid w; \theta)$$

This would be correct for an original sequence labeling task where it is important to find the correct labels (under 0–1 loss).

Here, however, sequence labeling arose from a transformation of the underlying chunking task. The loss/utility function of the overall task should be respected, which leads to maximum expected utility decoding.

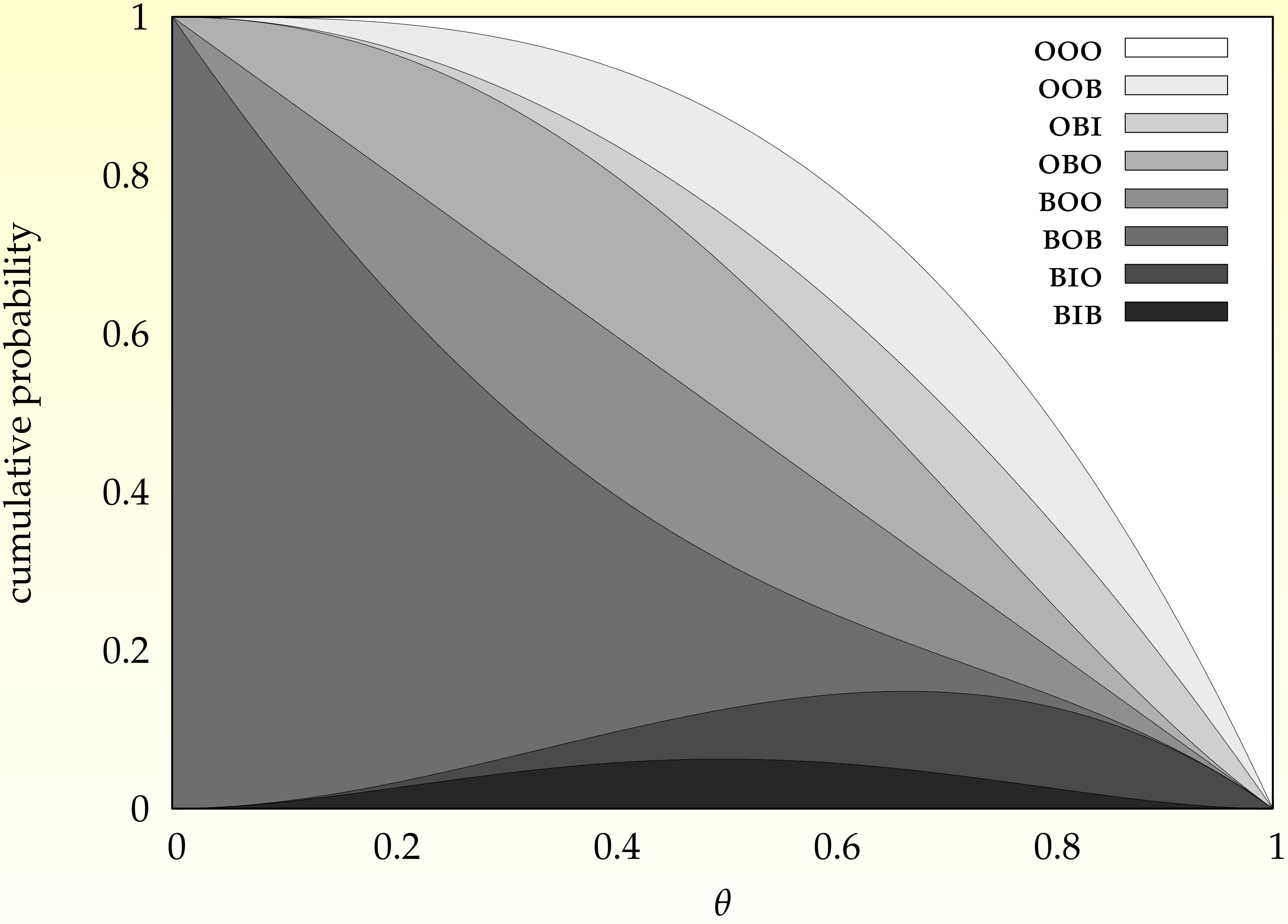
11 A toy example

Assume that $\Pr(y \mid w)$ is supplied by a bigram model over labels (which does not take word sequences into account at all):

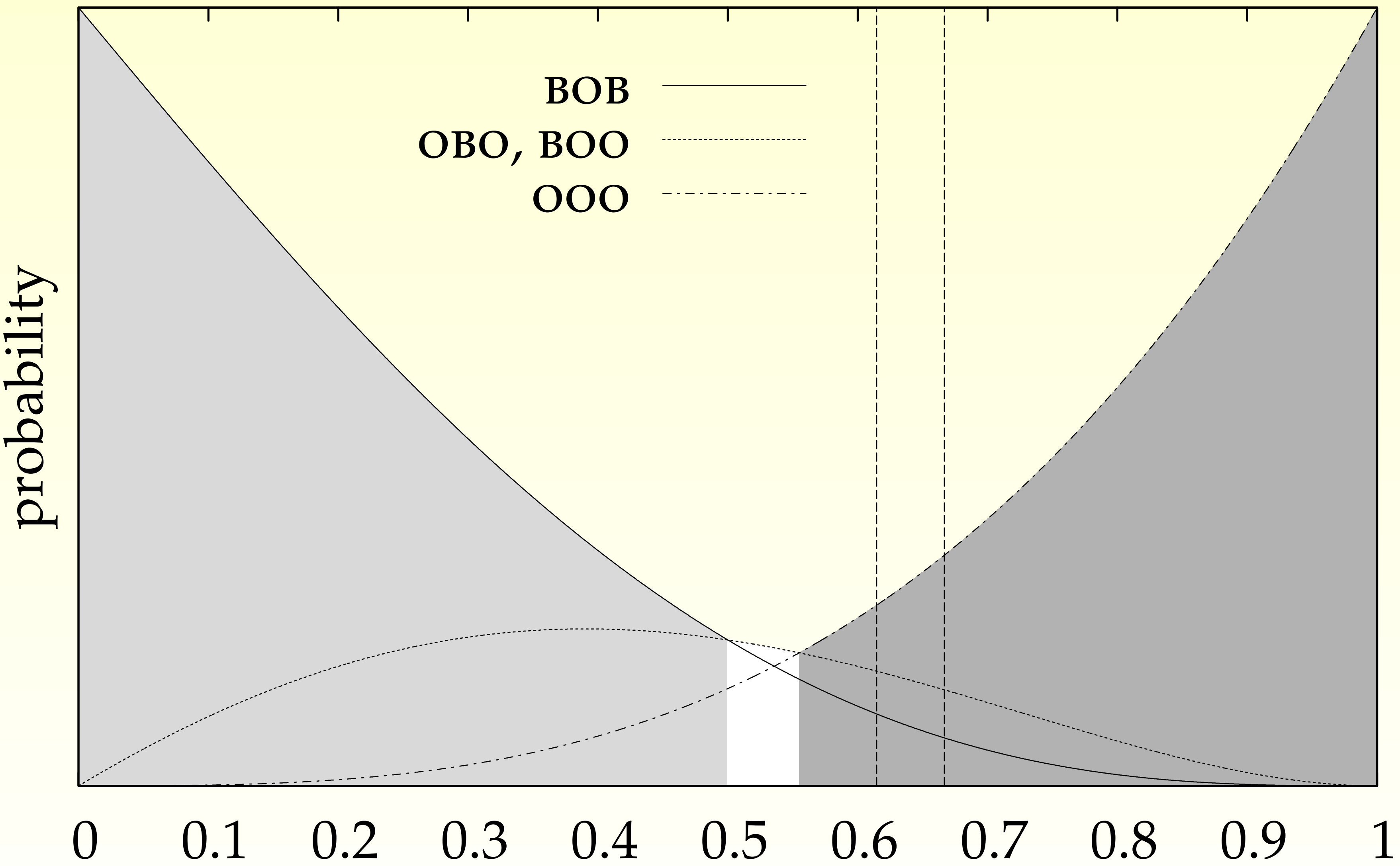
$$\Pr(y \mid w) = b(y_1 \mid \mathsf{O}; \theta) \prod_{i=2}^{|y|} b(y_i \mid y_{i-1}; \theta),$$

where b is as follows:

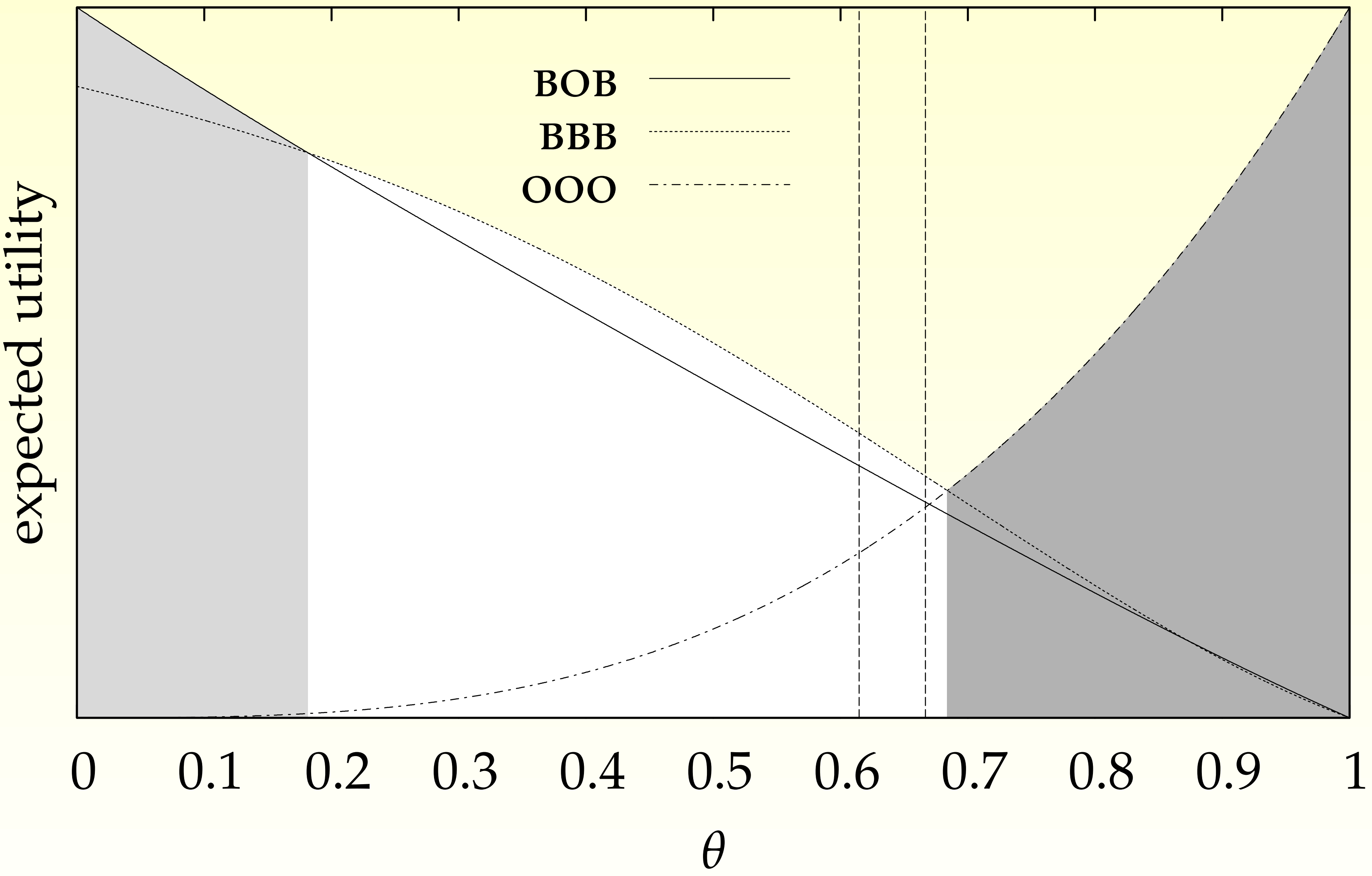
$$\begin{array}{ll}
b(\mathrm{I} \mid \mathrm{I}; \theta) = b(\mathrm{I} \mid \mathsf{O}; \theta) = 0 & b(\mathrm{I} \mid \mathsf{B}; \theta) = \theta^2 \\
b(\mathsf{O} \mid \mathrm{I}; \theta) = b(\mathsf{O} \mid \mathsf{O}; \theta) = \theta & b(\mathsf{O} \mid \mathsf{B}; \theta) = 1 - \theta^2 \\
b(\mathsf{B} \mid \mathrm{I}; \theta) = b(\mathsf{B} \mid \mathsf{O}; \theta) = 1 - \theta & b(\mathsf{B} \mid \mathsf{B}; \theta) = 0.
\end{array}$$


cumulative probability
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
θ
OOO
OOB
OBI
OBO
BOO
BOB
BIO
BIB

MAP decoding



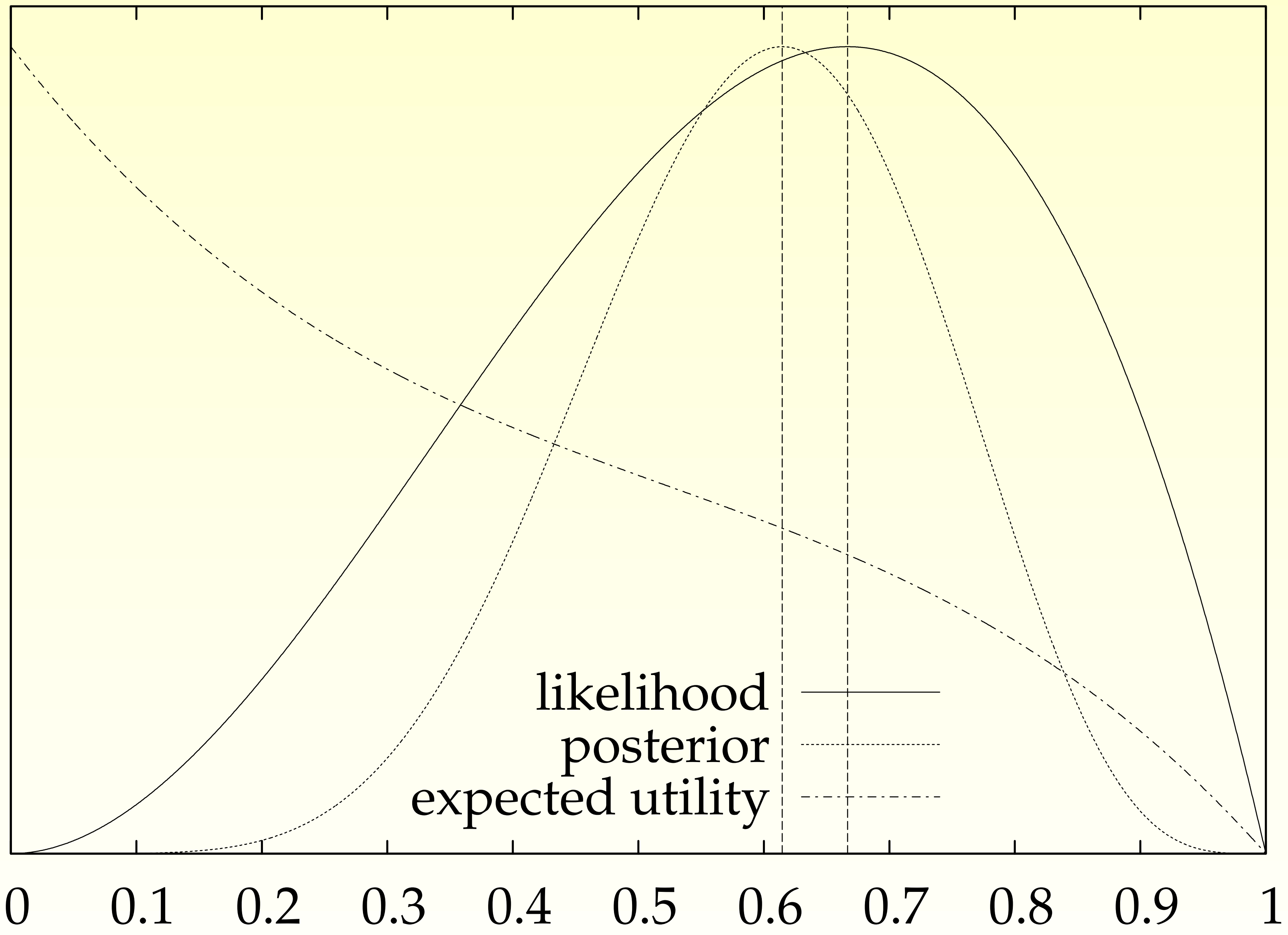
MEU decoding



Suppose the gold standard was OOB. How would the decoded hypotheses fare?

x	$P(x \mid \text{OOB})$	$R(x \mid \text{OOB})$	$F_3(x \mid \text{OOB})$
OOO	0/0	0/1	0.00
OBO	0/1	0/1	0.00
BOO	0/1	0/1	0.00
BBB	1/3	1/1	0.67
BOB	1/2	1/1	0.80
OOB	1/1	1/1	1.00

What value of θ should we pick when confronted with the training sample OOB?


likelihood
posterior
expected utility
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

12 The learning task

- Assume a partially specified probability model with unknown parameter vector θ.
- Given a word sequence w plus corresponding label sequence x, we want to estimate θ.
- Derive a point estimate of θ as

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \mathcal{U}(x \mid w; \theta)$$

In the NLP literature, the estimation step is replaced by

$$\tilde{\theta} = \operatorname*{argmax}_{\theta} \Pr(x \mid w; \theta)$$

This would be optimal for the isolated sequence learning problem under some additional assumptions.

Speech recognition appears to be the only allied field that has pursued minimum risk (= maximum expected utility) parameter estimation.

Foundational questions about the use of MEU estimation remain.

13 The algorithmic challenge

Recap:

$$\begin{aligned} \text{Decoding:} \quad & \operatorname*{argmax}_{x} \mathcal{U}(x \mid w; \theta) \\ \text{Estimation:} \quad & \operatorname*{argmax}_{\theta} \mathcal{U}(x \mid w; \theta) \end{aligned}$$

where

$$\mathcal{U}(x \mid w; \theta) \propto \sum_{y} \frac{tp(x, y)}{m(x) + \beta\, m(y)} \Pr(y \mid w; \theta)$$

At the very least, we need to be able to evaluate $\mathcal{U}(x \mid w; \theta)$ efficiently.

14 The solution in a nutshell

Express the relevant computations as weighted state machines. Combine algorithms from formal language and graph theory to compute expected utility.

- Many useful probability models (HMMs, CMMs) can be viewed as stochastic finite state machines.

- The computation of matching chunks (true positives) can be carried out by two-tape state machines.

- The counting of chunks can be carried out by one-tape state machines.

- The counting of chunks can be carried out by one-tape state machines.

Let S and T be two finite state transducers over a common alphabet. Then their composition $S \circ T$ carries out the following computation:

$$[S \circ T](x, z) = \sum_{y} [S](x, y) \times [T](y, z)$$

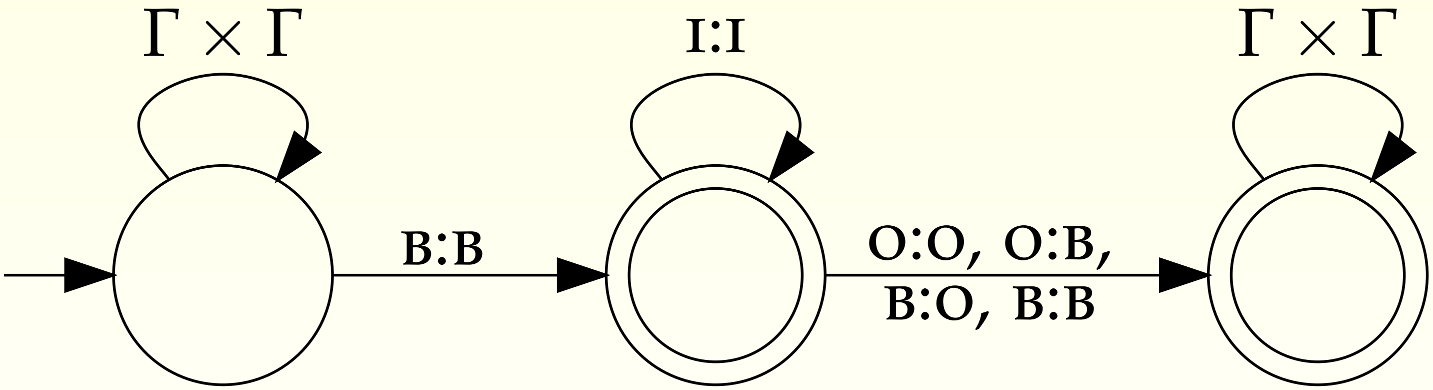
The expectations we want to compute are exactly of this form. They will be computed by composition of the elementary machines outlined above.

15 Counting matching chunks

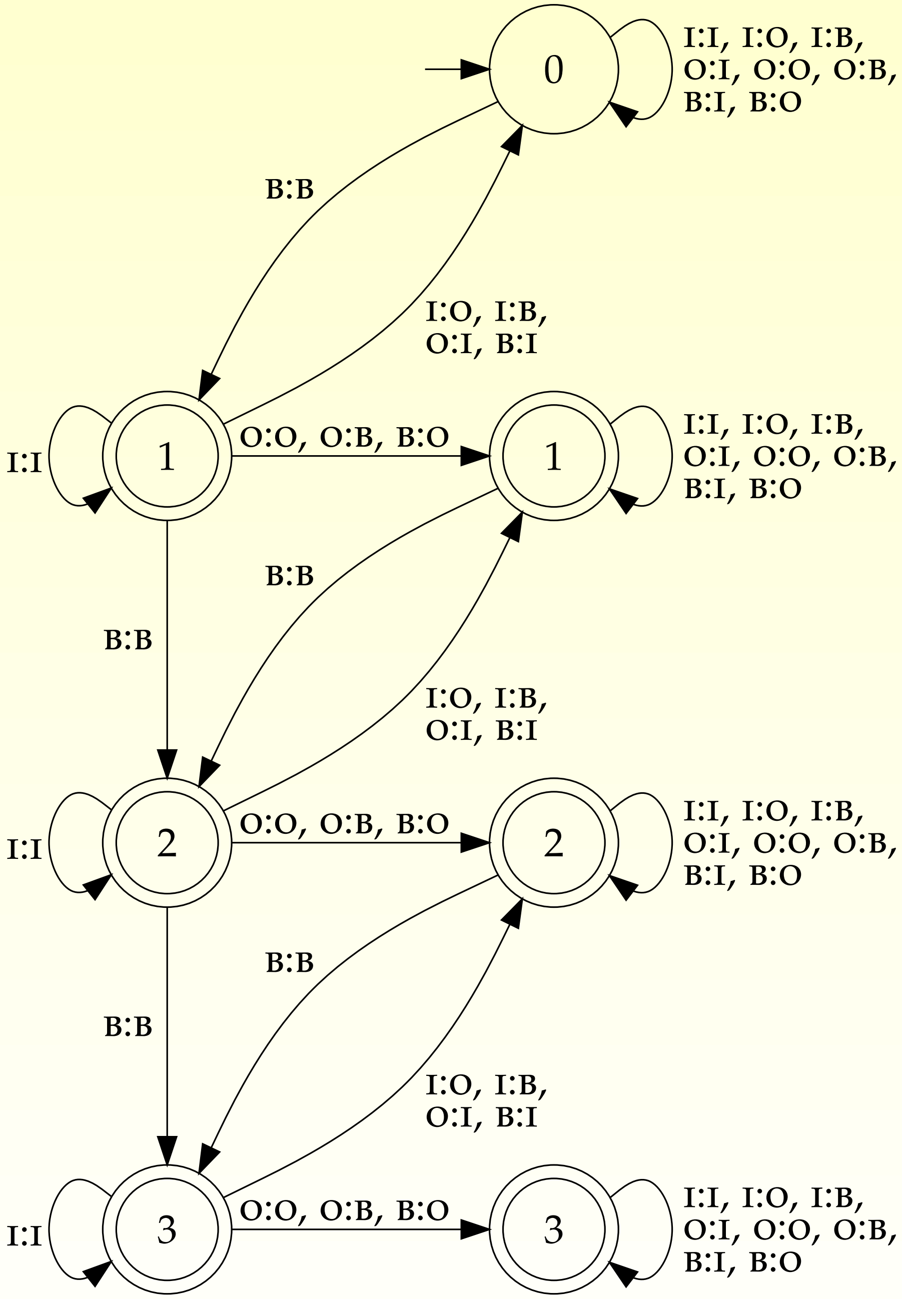
What constitutes a matching chunk?

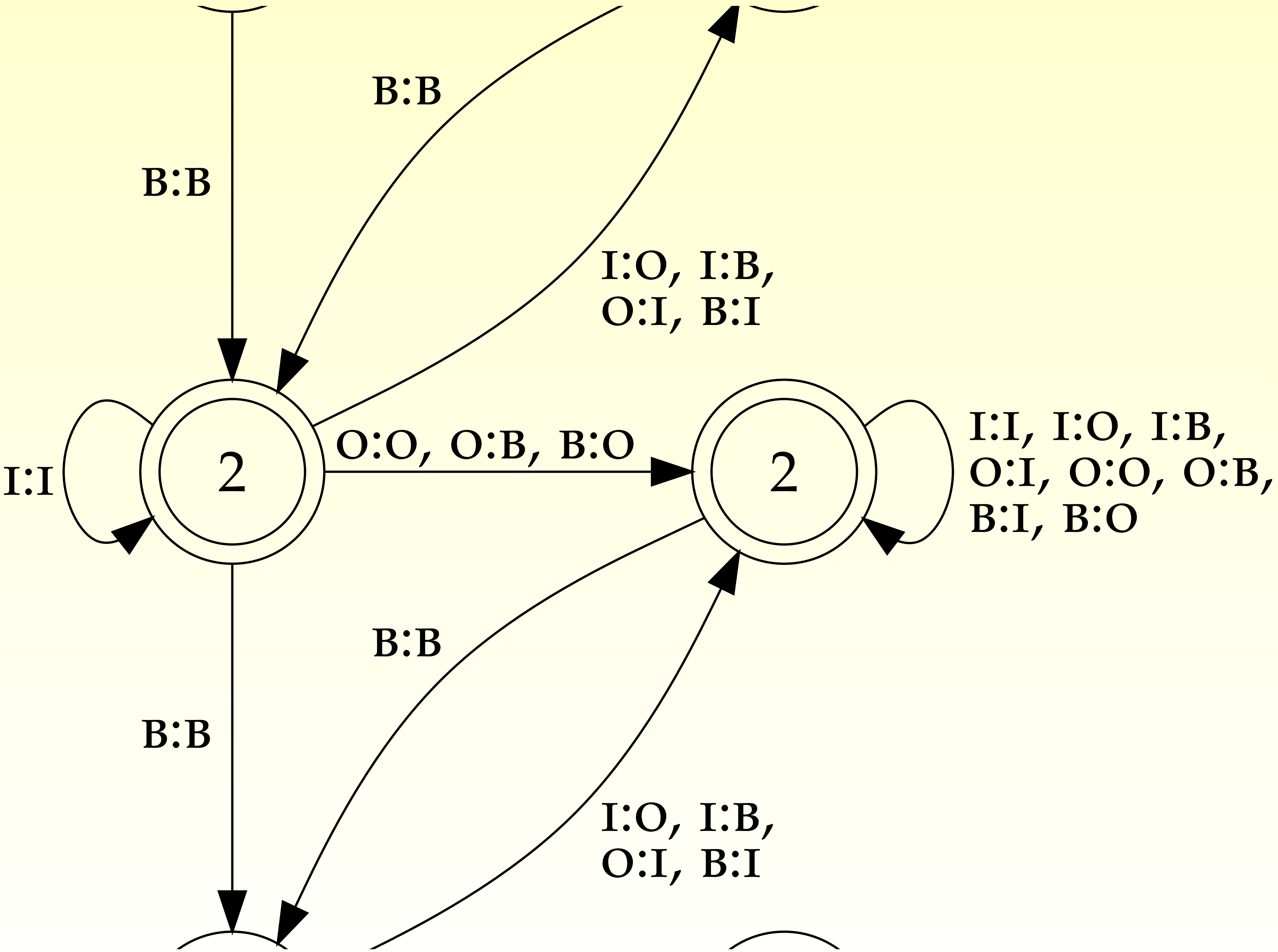
$$\mathsf{B} : \mathsf{B}\ (\,\mathrm{I}\ :\ \mathrm{I}\,)^*\ (\$: \$\,|\,\mathsf{O} : \mathsf{O}\,|\,\mathsf{O} : \mathsf{B}\,|\,\mathsf{B} : \mathsf{O}\,|\,\mathsf{B} : \mathsf{B})$$

Could use the counting technique from (Allauzen et al., 2003):



Nondeterminism is a bit problematic.


0
I:I, I:O, I:B,
O:I, O:O, O:B,
B:I, B:O
B:B
I:O, I:B,
O:I, B:I
I:I
1
O:O, O:B, B:O
1
I:I, I:O, I:B,
O:I, O:O, O:B,
B:I, B:O
B:B
B:B
I:O, I:B,
O:I, B:I
I:I
2
O:O, O:B, B:O
2
I:I, I:O, I:B,
O:I, O:O, O:B,
B:I, B:O
B:B
B:B
I:O, I:B,
O:I, B:I
I:I
3
O:O, O:B, B:O
3
I:I, I:O, I:B,
O:I, O:O, O:B,
B:I, B:O


B:B
B:B
I:O, I:B,
O:I, B:I
I:I
2
O:O, O:B, B:O
2
I:I, I:O, I:B,
O:I, O:O, O:B,
B:I, B:O
B:B
B:B
I:O, I:B,
O:I, B:I

An infinite state automaton T; deterministic when reading both tapes simultaneously.

A state is a tuple

$$(ct, match)$$

where ct is the current number of matching chunks (true positives) and $match$ is a Boolean variable indicating whether a potentially matching chunk is currently open.

Can compute $tp(x, y)$ as $[T](x, y)$. This requires expanding at most $(|x| + 1) \times (2\, tp(x, y) + 1)$ states, which is $O(|x|^2)$ in the worst case.

16 Computing expected precision

Expected precision:

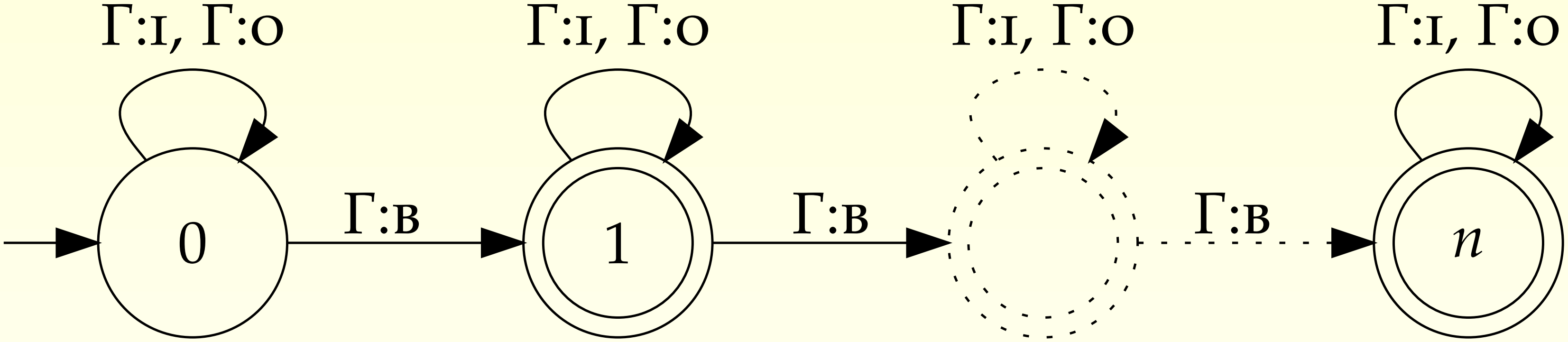
$$\mathsf{E}[P(x \mid \cdot)] = 1/m(x) \sum_y tp(x,y) \Pr(y \mid w)$$

Assume the probability model can be represented as a transducer Q. Then the expected number of true positives can be computed by $T \circ Q$:

$$\begin{aligned}[T \circ Q](x,w) &= \sum_y [T](x,y) \times [Q](y,w) \\ &= \sum_y tp(x,y) \Pr(y \mid w)\end{aligned}$$

17 Counting chunks

Simply count occurrences of the label B on the second tape, essentially ignoring the first tape:



Call this transducer M.

18 Computing expected recall

Expected recall:

$$\mathsf{E}[R(x \mid \cdot)] = \sum_y \frac{tp(x,y)}{m(y)} \Pr(y \mid w)$$

We need a composition-like operation, but defined on the codomain of transducers, rather than their domain. If A and B are transducers, let $A \boxtimes B$ have the following behavior:

$$[A \boxtimes B](x,y) = ([A](x,y), [B](x,y))$$

$A \boxtimes B$ can be constructed as the composition of transducers derived from A and B (without any increase in size).

Let T and M as defined before. Then $T \boxtimes M$ is essentially a transducer with states of the form

$$(ct, match, cm)$$

where ct and $match$ play the same role as in T and cm is the number of chunks seen so far on the second tape.

Abusing notation, we can say that $(T \boxtimes M) \circ Q$ computes the probabilities of all (ct, cm) combinations.

19 Computing expected utility

Expected F-measure:

$$\mathcal{U}(x \mid w; \theta) = (\beta + 1) \sum_{y} \frac{tp(x, y)}{m(x) + \beta\, m(y)} \Pr(y \mid w; \theta)$$

This computation is structurally the same as the computation of expected recall, since $m(x)$ and β are constants.

The number of states explored when the probability model is first-order Markov is at most $(|x| + 1) \times (2\, m(x) + 1) \times (|x| + 1) \times 3$, which is $O(|x|^3)$.

20 Parameter estimation

Under reasonable assumptions about the probability model, we can also express

$$\frac{\partial}{\partial \theta_j} \mathcal{U}(x \mid w; \theta)$$
$$= (\beta + 1) \sum_y \frac{tp(x, y)}{m(x) + \beta\, m(y)} \times \frac{\partial}{\partial \theta_j} \Pr(y \mid w; \theta)$$

in terms of state machines. This is sufficient for maximizing θ using conjugate gradient or similar multidimensional optimization algorithms.

21 Decoding

Contrast:

$$\begin{aligned} \text{Decoding:} \quad & \operatorname*{argmax}_{x} \mathcal{U}(x \mid w; \theta) \\ \text{Estimation:} \quad & \operatorname*{argmax}_{\theta} \mathcal{U}(x \mid w; \theta) \end{aligned}$$

Decoding appears to be more difficult than estimation, since it involves a combinatorial optimization step over exponentially many hypotheses x. Doing this naively is tractable in many practical cases.

22 Conclusion

- General lesson: Need to be careful not to lose track of the overall evaluation criterion when reducing a processing/learning problem to a more familiar one.
- For chunking, label sequence models need to be informed by the loss/utility function associated with the chunking task.
- Expected utility and its parameter gradient can be evaluated in cubic time. This makes MEU parameter estimation feasible.

23 Open problems

An NFA A is unambiguous if every string accepted by A is accepted by exactly one path through A.

Problem: Given an acyclic NFA A, find an equivalent unambiguous NFA B which is at most polynomially larger than A.

If this problem can be solved for the present case, efficient decoding is possible.

References

Cyril Allauzen, Mehryar Mohri, and Brian Roark. 2003. Generalized algorithms for constructing language models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 40–47. Sapporo, Japan. ACL Anthology P03-1006.

Yoshua Bengio. 1999. Markovian models for sequential data. *Neural Computing Surveys*, 2:129–162.

Richard O. Duda, Peter E. Hart, and David G.

Stork. 2000. Pattern Classification. Wiley, New York, second edition.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In Proceedings of the 17th International Conference on Machine Learning, pages 591–598. Stanford, CA.

Adwait Ratnaparkhi. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics, pages 173–179. Bergen, Norway. ACL Anthology E99-1023.

C. J. van Rijsbergen. 1974. Foundation of evaluation. Journal of Documentation, 30(4):365–373.