Efficient Minimum Error Rate Training and Minimum Bayes-Risk Decoding for Translation Hypergraphs and Lattices

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och

Presented By: Jeff Flanigan

April 18, 2012

Overview

- Hypergraph MERT
- Efficient Minimum Bayes-Risk (MBR) Decoding for Lattices
- MBR Decoding on Hypergraphs
- Combine all three
- Results

MERT Review

Decoder rule:

$$\hat{E}(F_s; \lambda_1^M) = \arg\max_E \left\{ \sum_{m=1}^{M} \lambda_m h_m(E, F_s) \right\}$$

MERT Tuning:

$$\hat{\lambda}_1^M = \arg\min_{\lambda_1^M} \left\{ \sum_{s=1}^{S} \mathrm{Err}\big(R_s, \hat{E}(F_s; \lambda_1^M)\big) \right\}$$

F_s: source
R_s: reference translation
h_m: feature functions
λ_m: feature weights

MERT Review

Decoder rule: $$\hat{E}(F_s;\lambda_1^M) \quad = \quad \arg\max_{E} \left\{ \sum_{m=1}^{M} \lambda_m h_m(E, F_s) \right\}$$

Choose direction d_1^M $$\gamma \;= \lambda_1^M + \gamma \cdot d_1^M$$

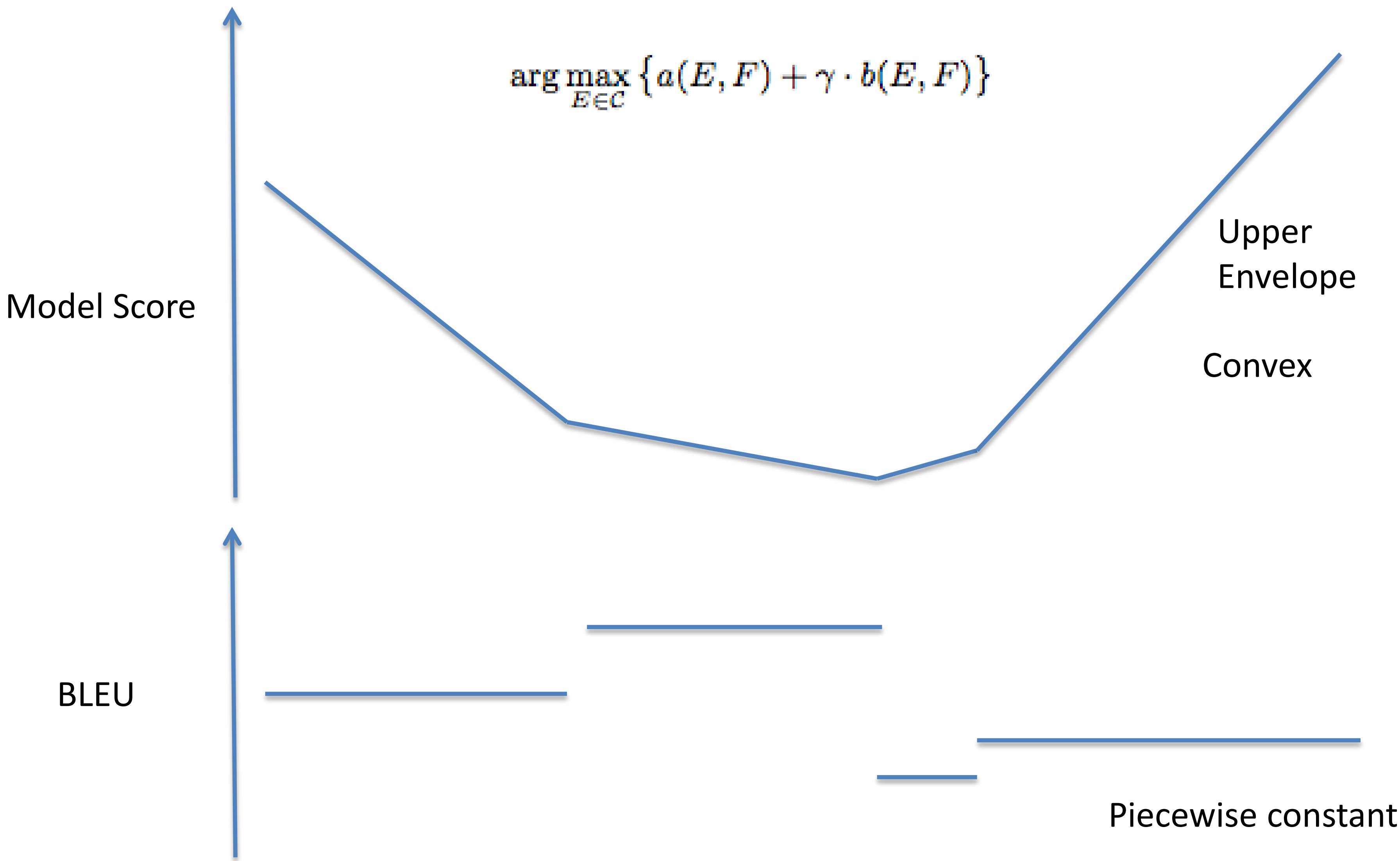
$$\hat{E}(F;\gamma) = \arg\max_{E \in \mathcal{C}} \left\{ (\lambda_1^M + \gamma \cdot d_1^M)^\top \cdot h_1^M(E,F) \right\}$$

$$= \arg\max_{E \in \mathcal{C}} \left\{ \underbrace{\sum_m \lambda_m h_m(E,F)}_{=a(E,F)} + \gamma \cdot \underbrace{\sum_m d_m h_m(E,F)}_{=b(E,F)} \right\}$$

$$= \arg\max_{E \in \mathcal{C}} \left\{ a(E,F) + \gamma \cdot b(E,F) \right\}$$

MERT Review

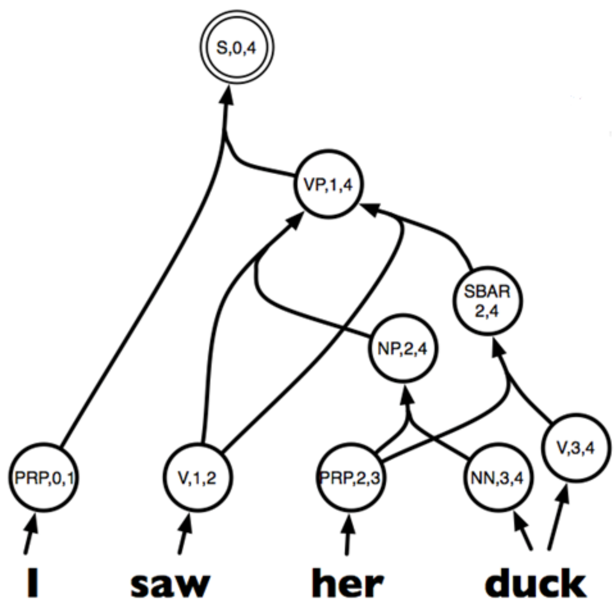
$$\arg\max_{E \in C} \{a(E,F) + \gamma \cdot b(E,F)\}$$



Hypergraphs

Hypergraph has nodes and edges H = <V,E>

Edges directed: go from multiple tails to single head



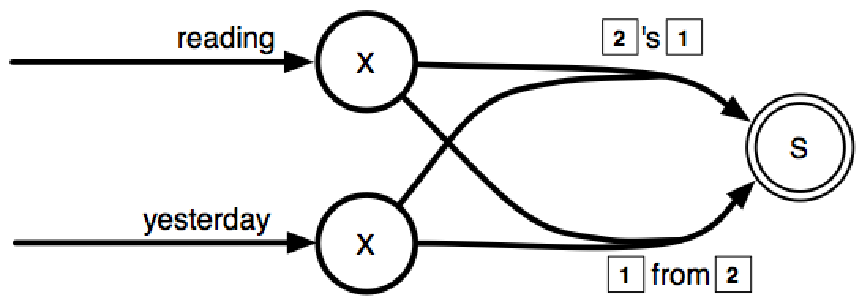
Source: http://www.cs.cmu.edu/afs/cs/project/cmt-55/lti/Courses/731/www/Spring-11/ClassSchedule2011.htm Chris Dyer's Slides 3/23

Translation Hypergraphs

Each edge labeled with a rule

Nodes = nonterminals

Path = translation hypothesis



$\{(\textit{yesterday 's reading}),$
$(\textit{reading from yesterday})\}$

Hypergraph MERT

Repeat until convergence:

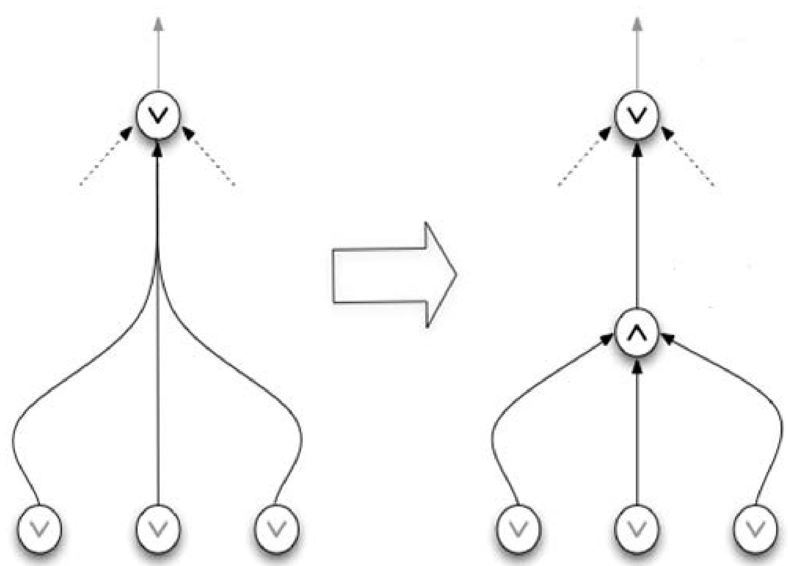
- Pick a direction
- Efficiently calculate upper envelope over entire lattice
- Line search for best BLEU score (for entire devset)

Hypergraph MERT

<u>Computing the upper envelope</u>

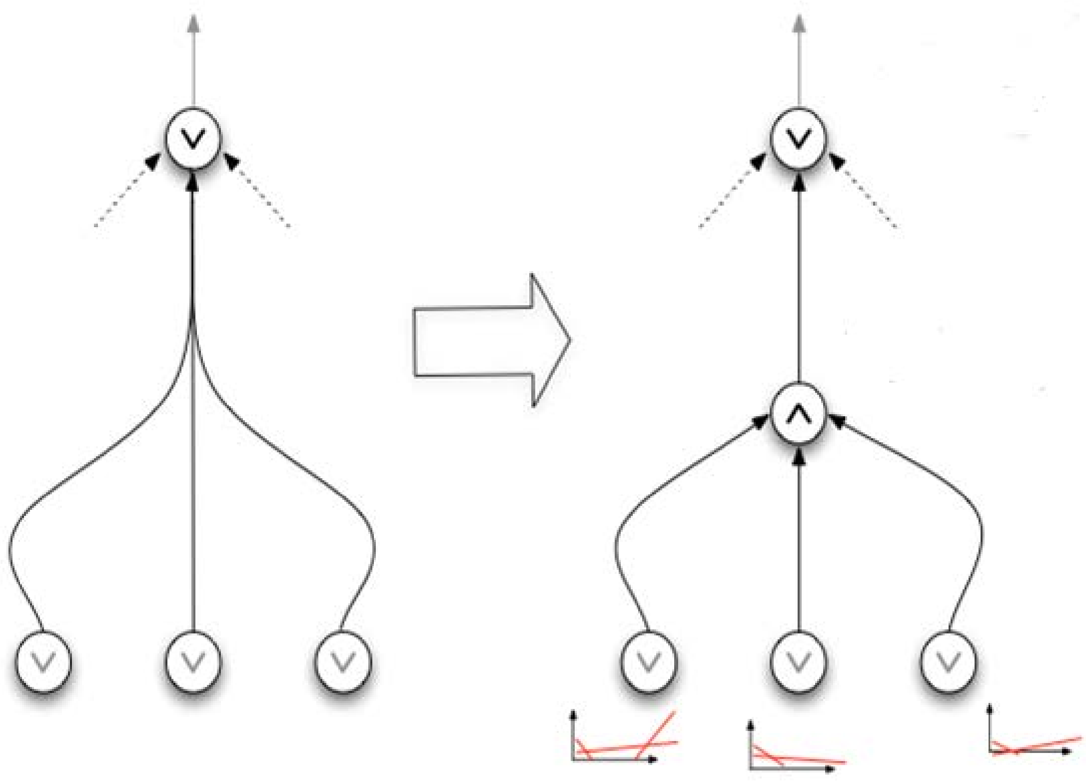
Step 1: Convert hypergraph to regular graph

- Nodes -> $\vee$ type nodes
- Edges -> $\wedge$ type nodes



Hypergraph MERT

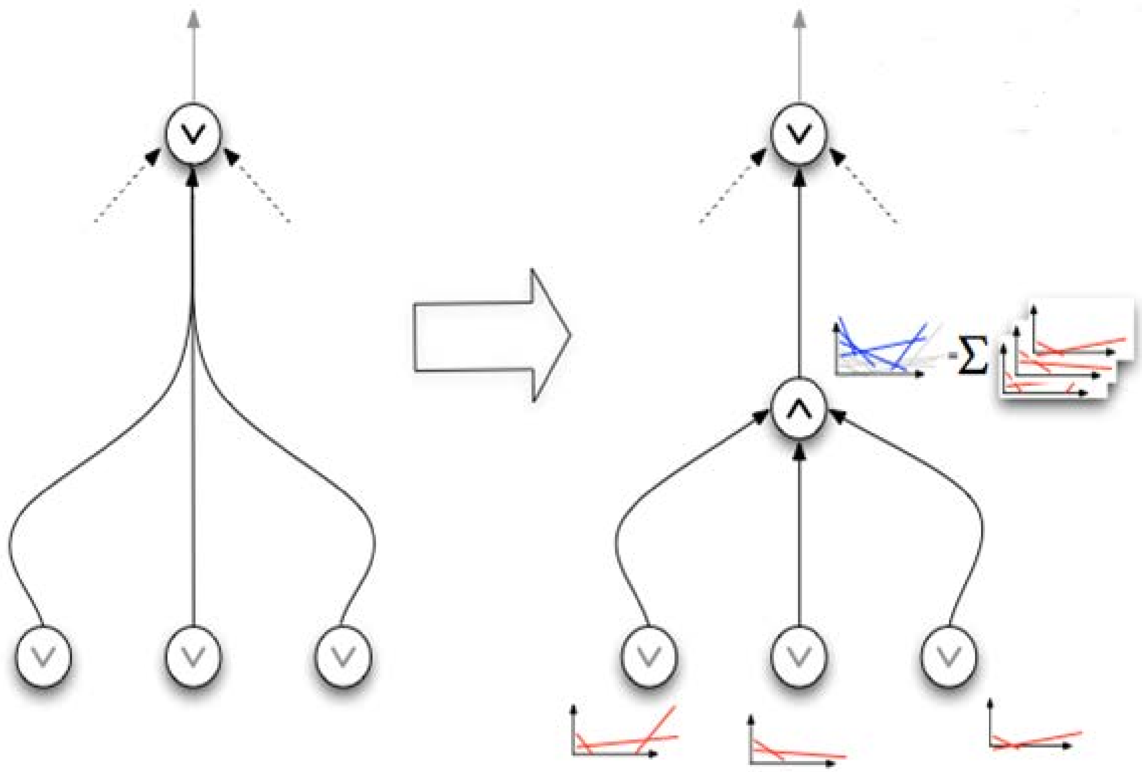
Step 2: Propagate upper envelope up to the root



Hypergraph MERT

Step 2: Propagate upper envelope up to the root

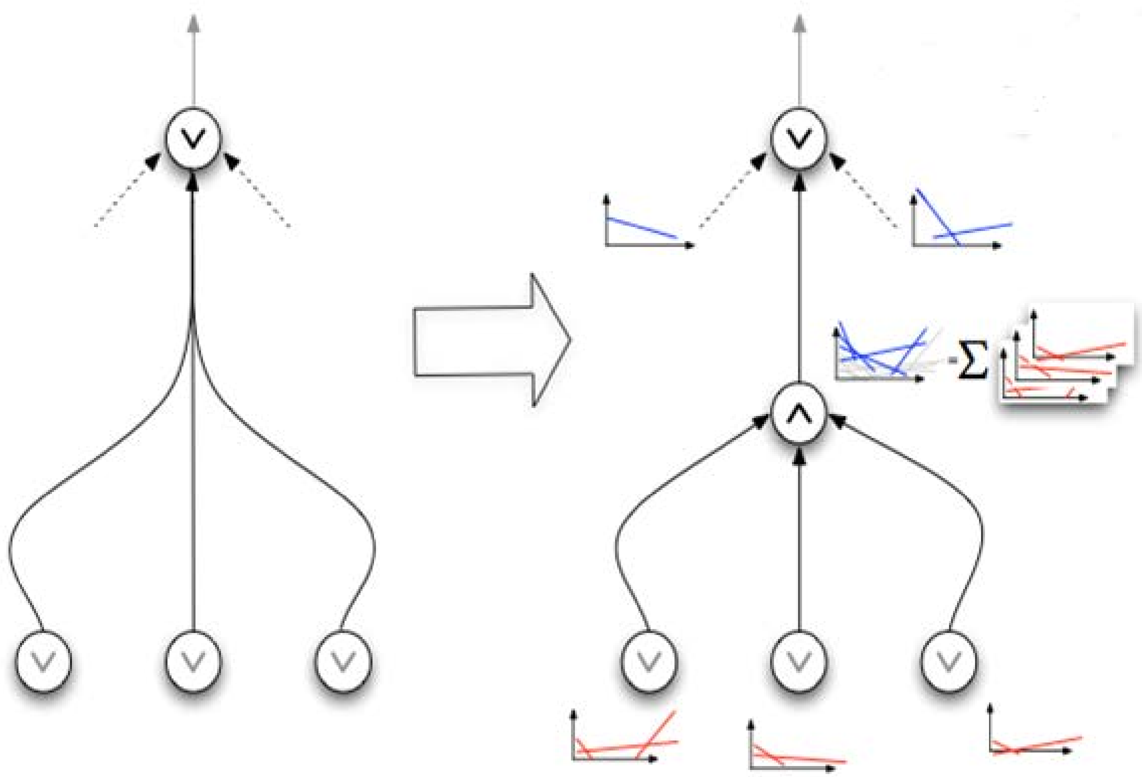
$\wedge$ nodes: Sum (also include rule score)



Hypergraph MERT

Step 2: Propagate upper envelope up to the root

$\wedge$ nodes: Sum (also include rule score)

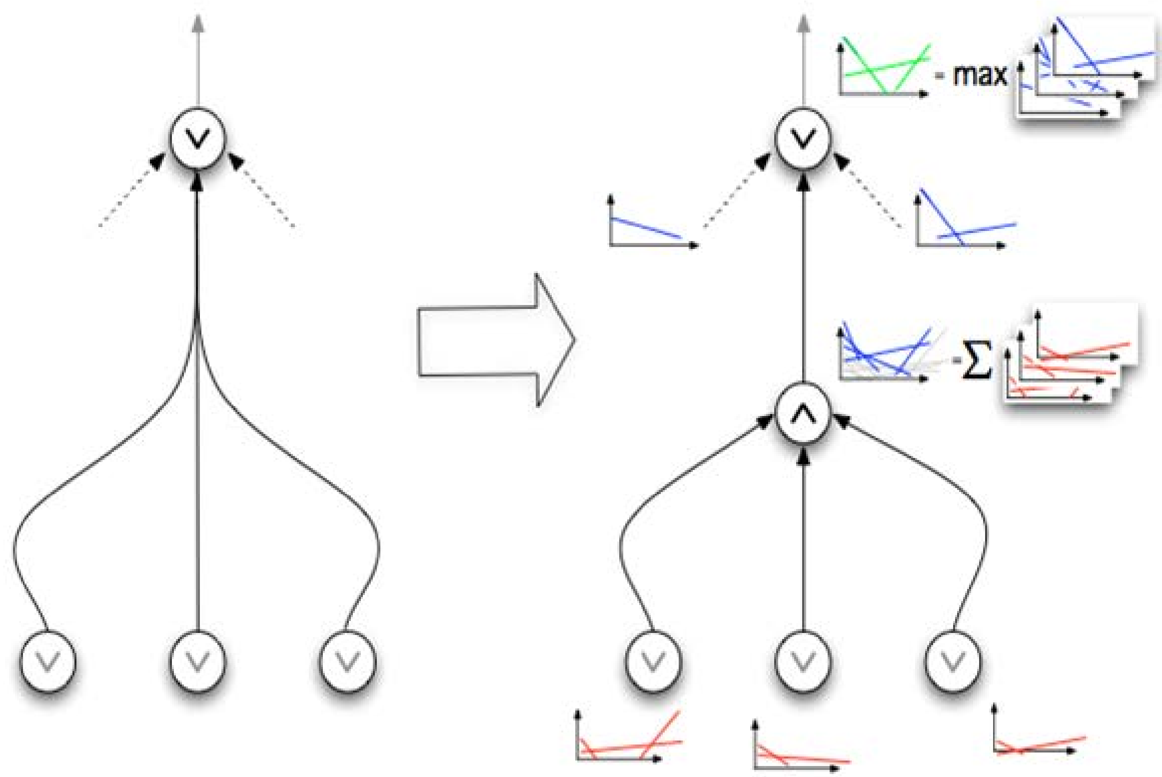


Hypergraph MERT

Step 2: Propagate upper envelope up to the root

$\wedge$ nodes: Sum (also include rule score)

$\vee$ nodes: Max



Minimum Bayes-Risk Decoding

Decoding Rule: $$\hat{E} = \underset{E' \in \mathcal{G}}{\operatorname{argmin}} \sum_{E \in \mathcal{G}} L(E, E') P(E|F)$$

Minimize expected loss under probability model P(E|F)

Minimum Bayes-Risk Decoding

Decoding Rule:

$$\hat{E} = \operatorname*{argmin}_{E' \in \mathcal{G}} \sum_{E \in \mathcal{G}} L(E, E') P(E|F)$$

-Loss = log(BLEU) ≈

$$G(E, E') = \theta_0 |E'| + \sum_{w} \theta_{|w|} \#_w(E') \delta_w(E)$$

1 if w is in E
0 otherwise

# of times w is in E’

Minimum Bayes-Risk Decoding

Decoding Rule: $$\hat{E} = \operatorname*{argmin}_{E' \in \mathcal{G}} \sum_{E \in \mathcal{G}} L(E, E') P(E|F)$$

1 if w is in E
0 otherwise

-Loss = log(BLEU) ≈ $$G(E, E') = \theta_0 |E'| + \sum_w \theta_{|w|} \#_w(E') \delta_w(E)$$

Sum over E
P(E|F) = 1

\# of times w is in E'

Plug in ⇒ $$\hat{E} = \operatorname*{argmax}_{E' \in \mathcal{G}} \theta_0 |E'| + \sum_w \theta_{|w|} \#_w(E') p(w|\mathcal{G})$$

where $$p(w|\mathcal{G}) = \sum_{E \in \mathcal{G}} 1_w(E) P(E|F)$$

"Posterior probability of n-gram w in the lattice G"

Efficient MBR on Lattices

Rewrite
$$p(w|\mathcal{G}) = \sum_{E\in\mathcal{G}} 1_w(E)P(E|F)$$

$$p(w|\mathcal{G}) \;=\; \sum_{E\in\mathcal{G}}\sum_{e\in E} f(e,w,E)P(E|F)$$

where
$$f(e,w,E) = \begin{cases} 1 & w \in e, p(e|\mathcal{G}) > p(e'|\mathcal{G}), \\ & e' \text{ precedes } e \text{ on } E \\ 0 & \text{otherwise} \end{cases}$$

Count w only once in hypothesis

Approximate f with f* = indicates edge containing w that has highest arc probability p(e|G)

f* can be calculated independent of path <= efficient

$$\begin{aligned} p(w|\mathcal{G}) &= \sum_{E\in\mathcal{G}}\sum_{e\in E} f^*(e,w,\mathcal{G})P(E|F) &&(7)\\ &= \sum_{e\in\mathcal{E}} 1_{w\in e} f^*(e,w,\mathcal{G}) \sum_{E\in\mathcal{G}} 1_E(e)P(E|F) \\ &= \sum_{e\in\mathcal{E}} 1_{w\in e} f^*(e,w,\mathcal{G}) P(e|\mathcal{G}), \end{aligned}$$

e ranges over all edges, pull out of sum

$P(e|\mathcal{G})$ is posterior prob of lattice edge

Efficient MBR on Lattices

Algorithm 3 MBR Decoding on Lattices

Sort the lattice nodes topologically.
Compute backward probabilities of each node.
Compute posterior prob. of each n-gram:
for each edge e **do**
    Compute edge posterior probability $P(e|\mathcal{G})$.
    Compute n-gram posterior probs. $P(w|\mathcal{G})$:
    for each n-gram w introduced by e **do**
        Propagate $n-1$ gram suffix to h_e.
        if $p(e|\mathcal{G}) > \text{Score}(w, T(e))$ **then**
            Update posterior probs. and scores:
            $p(w|\mathcal{G}) \mathrel{+}= p(e|\mathcal{G}) - \text{Score}(w, T(e))$.
            $\text{Score}(w, h_e) = p(e|\mathcal{G})$.
        else
            $\text{Score}(w, h_e) = \text{Score}(w, T(e))$.
        end if
    end for
end for
Assign scores to edges (given by Equation 3).
Find best path in the lattice (Equation 3).

Score(w,t) is highest probability of paths that terminate on t and contain n-gram w

Efficient MBR on Hypergraphs

Algorithm 4 MBR Decoding on Hypergraphs

Sort the hypergraph nodes topologically.
Compute inside probabilities of each node.
Compute posterior prob. of each hyperedge $P(e|\mathcal{G})$.
Compute posterior prob. of each n-gram:
for each hyperedge e **do**
    Merge the n-grams on the tail nodes $T(e)$. If the same n-gram is present on multiple tail nodes, keep the highest score.
    Apply the rule on e to the n-grams on $T(e)$.
    Propagate $n-1$ gram prefixes/suffixes to h_e.
    for each n-gram w introduced by this hyperedge **do**
        if $p(e|\mathcal{G}) > \text{Score}(w, T(e))$ **then**
            $p(w|\mathcal{G}) \mathrel{+}= p(e|\mathcal{G}) - \text{Score}(w, T(e))$
            $\text{Score}(w, h_e) = p(e|\mathcal{G})$
        else
            $\text{Score}(w, h_e) = \text{Score}(w, T(e))$
        end if
    end for
end for
Assign scores to hyperedges (Equation 3).
Find best path in the hypergraph (Equation 3).

Essentially the same as for lattices. Need to propagate n-gram prefixes and suffices

MERT for MBR Parameter Tuning

Use MERT to tune θ_i's

$$\hat{E} = \operatorname*{argmax}_{E' \in \mathcal{G}} \theta_0 |E'| + \sum_w \theta_{|w|} \#_w(E') p(w|\mathcal{G})$$

Also include additional feature $g_{N+1}(E,F)$ = original decoder cost (MAP translation)

Evaluation

NIST training data

Dataset	# of sentences	
	aren	zhen
dev	1797	1664
nist02	1043	878
nist03	663	919

	Avg. Runtime/sent [msec]			
	(Macherey 2008)		Suggested Alg.	
	aren	zhen	aren	zhen
phrase lattice	8.57	7.91	10.30	8.65
hypergraph	–	–	8.19	8.11

Table 2: Average time for computing envelopes.

Pruned hypergraph comparable running time

	BLEU (%)				Avg.
	aren		zhen		time
	nist03	nist02	nist03	nist02	(ms.)
MAP	54.2	64.2	40.1	39.0	-
N-best MBR	54.3	64.5	40.2	39.2	3.7
Lattice MBR					
FSAMBR	54.9	65.2	40.6	39.5	3.7
LatMBR	54.8	65.2	40.7	39.4	0.2

Table 3: Lattice MBR for a phrase-based system.

	BLEU (%)				Avg.
	aren		zhen		time
	nist03	nist02	nist03	nist02	(ms.)
Hiero					
MAP	52.8	62.9	41.0	39.8	-
N-best MBR	53.2	63.0	41.0	40.1	3.7
HGMBR	53.3	63.1	41.0	40.2	0.5
SAMT					
MAP	53.4	63.9	41.3	40.3	-
N-best MBR	53.8	64.3	41.7	41.1	3.7
HGMBR	54.0	64.6	41.8	41.1	0.5

Table 4: Hypergraph MBR for Hiero/SAMT systems.

Lattice MBR 20x faster than FSAMBR

Hypergraph MBR 7x faster than N-best MBR

Evaluation

System	BLEU (%)			
	MAP	MBR		
		default	mert-b	mert+b
aren.pb	54.2	54.8	54.8	54.9
aren.hier	52.8	53.3	53.5	53.7
aren.samt	53.4	54.0	54.4	54.0
zhen.pb	40.1	40.7	40.7	40.9
zhen.hier	41.0	41.0	41.0	41.0
zhen.samt	41.3	41.8	41.6	41.7

Table 5: MBR Parameter Tuning on NIST systems

Not much gain over default parameter settings

MBR wrt. MAP	default	mert-b	mert+b
# of gains	18	22	26
# of no-changes	9	5	8
# of drops	12	12	5

Table 6: MBR on Multi-language systems.

Gain over default parameter settings