0. Intro

Hypergraph View: Computation graph = labeled acyclic **hypergraph**. Edges can have multiple sources/targets. **Complexity**: same time as f; space higher (store intermediates) vec-vec: $O(d)$; mat-vec: $O(nm)$; mat-mat: $O(nm\ell)$

NLL ∇ = 0: $\sum_{i=1}^{n} \boldsymbol{f}(x_i, y_i) = \sum_{i=1}^{n} \mathbb{E}_{y|x_i,\theta}[\boldsymbol{f}(x_i, y)]$ Observed features = Expected features **Hessian**: $\boldsymbol{H} = \sum_i \mathrm{Cov}_{y|x_i,\theta}[\boldsymbol{f}(x_i, y)]$ (PSD!)

DAG Properties: Topological order 唯一确定; DP 子问题独立拆分可行; Gradient 反向传播良定义(no cycles) **Hypergraph**: 函数式计算自然表示, multi-inputs→one output

1. Backpropagation

Chain: $\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$ **Jacobian**: $f: \mathbb{R}^n \to \mathbb{R}^m$, $\frac{dy}{dx} = \left[\frac{dy}{dx_1}, \ldots, \frac{dy}{dx_n}\right] \in \mathbb{R}^{m\times n}$ **Multivar**: $\frac{dy_i}{dx_j} = \sum_{k=1}^{m} \frac{dy_i}{dz_k}\frac{dz_k}{dx_j}$

Bauer Path: $\frac{dy_i}{dx_j} = \sum_{p\in\mathcal{P}(j,i)} \prod_{(k,l)\in p} \frac{dz_l}{dz_k}$ $\mathcal{P}(j,i)$=all paths $j \to i$; worst $O(m^n)$, m平均出度, n路径长度

Forward vs Reverse: **Forward**: expand $\frac{d}{dx}$ recursively, same flow as fwd **Reverse**: 2 passes—fwd compute vals, bwd compute grads **Complexity**: same time as f; higher space (store intermediates)

Primitives: **Sum**: $\frac{d(a+b)}{da} = 1$; **Prod**: $\frac{d(ab)}{da} = b$

2. Log-Linear Models

Prob Basics: **Bayes**: $p(y|x) = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy}$ Posterior $\propto$ Prior $\times$ Likelihood **Marginal**: $p(x) = \sum_y p(x, y)$ **Expectation**: $\mathbb{E}[f(x)] = \sum_x f(x)p(x)$

Log-Linear Model: $p(y|x,\theta) = \frac{\exp(\theta\cdot f(x,y))}{Z(\theta)}$ $Z(\theta) = \sum_{y'\in Y} \exp(\theta \cdot f(x, y'))$ $\log p(y|x,\theta) = \theta \cdot f(x,y) - \log Z$ (linear in log space!) **Discrete MLE**: $p(y|x) = \frac{\mathrm{count}(x,y)}{\mathrm{count}(x)}$ (sparse 问题)

MLE ∇: $\theta_{\mathrm{MLE}} = \arg\min_\theta -\sum_{n=1}^{N} \log p(y_n|x_n, \theta)$ 观测特征 count = 期望特征 count → **Expectation Matching** $\frac{\partial\mathcal{L}}{\partial\theta_k} = -\sum_n f_k(x_n, y_n) + \sum_n \sum_{y'} p(y'|x_n;\theta) f_k(x_n, y')$

MAP & Ridge: $\hat{\theta}_{\mathrm{MAP}} = \arg\min[-\log p(\theta) - \log p(D|\theta)]$ Gaussian prior $\mathcal{N}(0, \sigma_p^2 I)$ → L2: $\frac{\lambda}{2}\|\theta\|^2$ Laplace prior → L1 regularization

Softmax: $\mathrm{softmax}(h, y, T) = \frac{\exp(h_y/T)}{\sum_{y'}\exp(h_{y'}/T)}$ $T \to 0$: argmax; $T \to \infty$: uniform $\log\mathrm{softmax} = h_y - \log\sum_{y'}\exp(h_{y'})$ (logsumexp)

MLP Architecture: **Problem**: Log-linear needs linearly separable data **Solution**: Learn non-linear feature fn $\boldsymbol{h}_k = \sigma_k(\boldsymbol{W}_k^\top \boldsymbol{h}_{k-1})$, $\boldsymbol{h}_1 = \sigma_1(\boldsymbol{W}_1^\top \boldsymbol{e}(x))$ Output: $\mathrm{softmax}(\boldsymbol{\theta}^\top \boldsymbol{h}_n)$

Activations: $\sigma(x) = \frac{1}{1+\exp(-x)}$, $\nabla\sigma = \sigma(1-\sigma)$ **tanh**: $\frac{1-e^{-2x}}{1+e^{-2x}}$, $\nabla = 1 - \tanh^2$ Sigmoid/tanh vanishing gradient→ use ReLU **Backprop**: $\frac{\partial\ell}{\partial \boldsymbol{W}_k} = \frac{\partial\ell}{\partial y}\frac{\partial y}{\partial h_n}\left(\prod_{m=k+1}^{n} \sigma'_m \boldsymbol{W}_m\right)\sigma'_k \boldsymbol{h}_{k-1}$

Exp Family & MaxEnt: $p(x|\theta) = \frac{1}{Z(\theta)}h(x)\exp(\theta\cdot\varphi(x))$ **Max Entropy**: $H(p) = -\sum_x p(x)\log p(x)$ 选最大熵分布=最少假设=Laplace 原则 **优势**: Conjugate priors; Sufficient stats; Convex log-partition → unique MLE

3. Language Models

Structured Prediction: **Kleene V^***: infinite set of finite-length strings from V **Language Model**: weighted prefix tree, each sentence=unique path $p(y) = \frac{1}{Z}\prod_{t=1}^{|y|} \mathrm{weight}_{y_{\le t}}$

Local Normalization: $Z = 1$ when children edges sum to 1 at each node **Consistency**: $p(\mathrm{EOS}|y_{<t}, V^*) > \varepsilon > 0$ $p(|y| = \infty) \le \lim_{t\to\infty}(1-\varepsilon)^t = 0$ (tight)

N-gram Model: $p(y_t|y_{<t}) = p(y_t|y_{t-1}, \ldots, y_{t-n+1})$ **Markov**: $P(t_i|t_{1:i-1}) = P(t_i|t_{i-1})$ (1st order) $= \frac{\exp(w_{y_t}\cdot h_t)}{\sum_{y'\in V}\exp(w_{y'}\cdot h_t)}$, $h_t \in \mathbb{R}^d$ **Bengio**: $h_t = f(e(\mathrm{hist}))$, $e(\mathrm{hist}) = [e(y_{t-1}); e(y_{t-2}); \ldots]$

RNN: $h_t = f(h_{t-1}, e(y_{t-1}))$ (implicit infinite context) **Vanilla**: $h_t = \sigma(W_1 h_{t-1} + W_2 e(y_{t-1}))$ **BPTT**: unroll through time, sum grads over timesteps

4. Word Embeddings

Encoding: **One-hot**: $v \in O(|V|)$, only word=1 **Bag-of-words**: pooled one-hot (sum/mean/max) **N-grams**: vectors huge—every combo needs slot **Pipeline**: Embedding → Pooling → NN → Softmax

Skip-gram: **Preprocess**: word-context pairs ($k \times C$ many), window k $p(c|w) = \frac{1}{Z(w)}\exp\left(e_{\mathrm{wrd}(w)} \cdot e_{\mathrm{ctx}(c)}\right)$, $O(2|V|k)$ params **Bilinear**: linear if all-but-one vars held constant **Similarity**: $\cos(u_i, u_j)$

5. CRF & POS Tagging

As Graph: Fully connected graph w/ POS-tag nodes per layer $\mathrm{score}(\langle D, N, V, \ldots\rangle, w) = \theta f(t, w)$ **score** (t, w)= unnormalized log-prob = $\sum_n$ trans+emit Problem: $O(|\mathcal{T}|^N)$ paths in normalizer

CRF Model: $p(t|w) = \frac{\exp(\mathrm{score}(t,w))}{\sum_{t'\in\mathcal{T}^N}\exp(\mathrm{score}(t',w))}$ **Decomposition**: $\mathrm{score}(t, w) = \sum_{n=1}^{N} \mathrm{score}(\langle t_{n-1}, t_n\rangle, w, n)$ $p(t|w) \propto \prod_{n=1}^{N} \exp\{\mathrm{score}(\langle t_{n-1}, t_n\rangle, w)\}$

DP 推导: $O(|\mathcal{T}|^N) \to O(N|\mathcal{T}|^2)$: Goal: $Z = \sum_{t\in\mathcal{T}^N} \exp \mathrm{score}(\boldsymbol{t}, \boldsymbol{w})$ **Step1**: 可加分解 $\mathrm{score} = \sum_n \mathrm{score}_n$ **Step2**: $Z = \sum_t \exp\sum_n \mathrm{score}_n = \sum_t \prod_n \exp\mathrm{score}_n$ (exp) **Step3**: $= \sum_{t_1}\cdots\sum_{t_N}\prod_n \exp\mathrm{score}_n$ (展开) **Step4**: $= \sum_{t_1}\exp\mathrm{score}_1 \times \left(\sum_{t_2}\cdots\right)$ (distrib 把内层 sum 推进去) **若 3-gram**: 依赖$t_{n-2}, t_{n-1}, t_n \to O(N|\mathcal{T}|^3)$

Forward Algorithm: $\alpha[0, t] = \exp(\mathrm{score}(\mathrm{BOS} \to t))$ (init w/ BOS trans) for $n = 1, \ldots, N-1$; for $t_n \in \mathcal{T}$: $\alpha[n, t_n] = \bigoplus_{t_{n-1}} \alpha[n-1, t_{n-1}] \otimes \exp(\mathrm{score})$ return $\bigoplus_t \alpha[N-1, t]$ (sum last column!) **直觉**: prefix 之和, 从 seq 开头走到当前状态的所有走法 score 总和

Backward Algorithm: $\forall t_N: \beta[N, t_N] \leftarrow 1$ for $n = N-1, \ldots, 0$; for $t_n \in \mathcal{T}$: $\beta[n, t_n] \leftarrow \bigoplus_{t_{n+1}} \exp(\mathrm{score}_{n+1}) \otimes \beta[n+1, t_{n+1}]$ return $\beta[0, \mathrm{BOS}]$ (single value!) **Complexity**: $O(N|\mathcal{T}|^2)$

Fwd vs Bwd Asymmetry: **Init**: Bwd 直接1; Fwd 需 BOS 转移 **Term**: Bwd 返回$\beta[0, \mathrm{BOS}]$单值; Fwd 需⊕整列 **原因**: BOS 显式存在, EOS 不显式处理

Viterbi Decoding: $\delta[n, t] = \max_{t_{n-1}}[\delta[n-1, t_{n-1}] + \mathrm{score}(t_{n-1}, t)]$ 每步枚举t和t_{n-1}→$|\mathcal{T}|^2$种 trans **Backtrack**: 存argmax指针 bp, 从$\arg\max_t \delta[N, t]$回溯

Common Semirings:

Name	$\mathbb{K}$	$\oplus$	$\otimes$	**0**	**1**	**用途**
Real	$\mathbb{R}_{\ge 0}$	$+$	$\times$	0	1	Z partition
Viterbi	$\mathbb{R}\cup\{-\infty\}$	max	$+$	$-\infty$	0	最优 path
Log	$\mathbb{R}\cup\{\pm\infty\}$	lse	$+$	$-\infty$	0	$\log Z$
Boolean	$\{0, 1\}$	$\vee$	$\wedge$	0	1	可达性
Counting	$\mathbb{N}$	$+$	$\times$	0	1	路径数
Tropical	$\mathbb{R}\cup\{\infty\}$	min	$+$	∞	0	最短路

Semiring Definition: $\langle\mathbb{K}, \oplus, \otimes, \mathbf{0}, \mathbf{1}\rangle$ where:
1. $(\mathbb{K}, \oplus, \mathbf{0})$: **comm monoid** (assoc+comm+identity)
2. $(\mathbb{K}, \otimes, \mathbf{1})$: **monoid** (assoc+identity)
3. **Distrib**: $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$
4. **Annihilator**: $\mathbf{0} \otimes x = x \otimes \mathbf{0} = \mathbf{0}$

陷阱: $\mathbf{0} = \mathbf{1}$必失败!

Semiring 意义: $\oplus$: **分治** (split points 合并, OR/MAX/+) $\otimes$: **连接** (左右子树组合, AND/×/+) **0**: 吸收元, 消除 invalid; **1**: 单位元, null 不破坏

Monoid 判定:
1. **Closure**: $a \otimes b \in \mathbb{K}$
2. **Assoc**: $(a \otimes b) \otimes c = a \otimes (b \otimes c)$
3. **Identity**: $\exists e: a \otimes e = e \otimes a = a$

Kleene Star: $a^* = \bigoplus_{n=0}^{\infty} a^{\otimes n} = \mathbf{1} \oplus a \otimes a^*$ Real 上 $|a| < 1$: $a^* = \frac{1}{1-a}$ (geometric series) Tropical: $a^* = 0$ if $a \ge 0$ (正环不帮助) 用于 globally normalized LM

6. CFG Parsing

Constituents: Multi-word units as single unit **Tests**: Pronoun substitution, Clefting, Answer ellipsis Ambiguity: PP attachment, modifier scope

CFG Definition: $G = \langle\mathcal{N}, \mathcal{S}, \Sigma, \mathcal{R}\rangle$ Non-terminals, start symbol, terminals, production rules **CNF**: $N_1 \to N_2 N_3$ or $N \to a$; $O(4^N)$ trees (Catalan)

Weighted CFG: **Global**: $p(t) = \frac{1}{Z}\prod_{r\in t}\exp(\mathrm{score}(r))$ $Z = \sum_{t'\in\mathcal{T}}\prod_{r'}\exp(\mathrm{score}(r'))$ (可能∞!) **Probabilistic**: local norm $\sum_k p(\alpha_k|N) = 1$

CKY Chart索引: Position 在 **words 之间**: $0|w_1|1|w_2|2|\ldots|N$ Chart$[i, k, X]$: span $[i, k)$覆盖 $w_i, \ldots, w_{k-1}$ **长度**: $k - i$; **对角线**: $k - i = 1$ (单词) **Fill order**: 按 span 长度递增($\ell = 1, 2, \ldots, N$) 同一长度内任意顺序 (topo order 自由度) **Goal**: Chart$[0, N, S]$

CKY algo: $O(N^3|R|)$, needs CNF **Terminal**: $C[i, i+1, X] = \exp\mathrm{score}(X \to w_i)$ for $X \to w_i \in \mathcal{R}$ **Binary**: for span$= 2, \ldots, N$; for $i = 1, \ldots, N -$ span: $k \leftarrow i +$ span; for $j = i+1, \ldots, k-1$; for $X \to YZ \in \mathcal{R}$: $C[i, k, X] \oplus \exp\{\mathrm{score}\} \otimes C[i, j, Y] \otimes C[j, k, Z]$

CKY Chart 3×3 Example: Sentence: $w_1 w_2 w_3$

	1	2	3
0	$C[0,1]$	$C[0,2]$	$C[0,3]$←**goal**
1		$C[1,2]$	$C[1,3]$
2			$C[2,3]$

Fill: diag first, then by span length

7. Dependency Parsing

Dependency Tree: Directed spanning tree, root degree 1 **Constraints**: Single head; Connected; Acyclic **Projective**: arcs 不交叉 (嵌套/并列) → CKY 可用 **Non-projective**: arcs 可交叉 → 必须用 CLE/MTT # spanning trees: $O((n-1)^{n-2})$

Edge-Factored Model: **优点**: 全局优化分解为局部边决策 **局限**: 无法捕捉 sibling/grandparent effects $\mathrm{score}(t, \boldsymbol{w}) = \sum_{(i\to j)\in t}\mathrm{score}(i \to j, \boldsymbol{w}) + \mathrm{score}(r, \boldsymbol{w})$ $p(t|w) = \frac{1}{Z}\prod_{(i\to j)\in t}\exp(\mathrm{score}(i, j, w))\exp(\mathrm{score}(r, w))$

Cayley Formula: **无向K_n**: n^{n-2}棵 spanning trees **有向+固定 root**: n^{n-2}棵 arborescences **有向+任意 root**: $n \times n^{n-2} = n^{n-1}$棵

Graph Laplacian L: $L_{ij} = \begin{cases}\mathrm{Degree}(i) & i=j \text{ (对角线)}\\ -1 & i\neq j \wedge i\sim j \text{ (有边)}\\ 0 & \text{otherwise}\end{cases}$

trick: 只看非对角−1判断边存在 **MTT**: #spanning trees = $\det(\hat{L})$ (any minor)

Weighted Laplacian (MTT): $A_{ij} = \exp(\mathrm{score}(i \to j))$, $\rho_j = \exp(\mathrm{score}(j, w))$ $L_{ij} = \begin{cases}\rho_j & i=1 \text{ (root row)}\\ \sum_{k\neq j} A_{kj} & i=j \text{ (in-degree)}\\ -A_{ij} & \text{else}\end{cases}$ $Z = \det(L)$, Computing in $O(n^3)$

CLE Algorithm: **Goal**: max spanning arborescence (directed MST)
1. For each node v, pick max incoming edge
2. If no cycle → done (it's a tree)
3. If cycle → **contract** cycle to supernode
4. **Reweight**: $\omega'(u \to v) = \omega(u \to v) - \omega_{\text{in-cycle}(v)}$
5. Recursively solve contracted graph
6. **Expand**: break cycle at min-loss edge

Complexity: $O(N^2)$ or $O(E + N\log N)$

Root Constraint: CLE base 允许多 root outgoing arcs **Naive**: 对每条 root arc 分别运行 CLE → $O(N \cdot \mathrm{CLE})$ **Clever** (Gabow): swap score=next-best - current 删除 swap score 最小的多余 root edge

MTT vs CLE: [**维度**], [**MTT**], [**CLE**], [目标], [$Z = \sum_t \exp(\mathrm{score})$], [$t^* = \arg\max$], [算法], [$\det(\hat{L})$], [Greedy+Contract], [复杂度], [$O(N^3)$], [$O(N^2)$],

8. Semantic Parsing

Syntax vs Semantics: **Syntax**: structural org (parse tree) **Semantics**: underlying meaning **Logical form**: quantifiers, vars, boolean, predicates **Compositionality**: meaning of whole = fn of parts

Lambda Calculus: **Terms**: 变量x; 抽象$\lambda x.M$; 应用(MN) **β-reduction**: $(\lambda x.M)N \to_\beta M[x := N]$ **α-conversion**: 重命名 bound 变量避免 capture **β-infinity**: $F = \lambda x(xx)x)$, $FF = \ldots$不终止

β-reduction 步骤:
1. 找到$(\lambda x.M)N$形式的 redex
2. 在M中找所有被该λx绑定的x
3. 将这些x替换为N

注意: 可能需先α-convert 避免变量捕获!

Free vs Bound Variables: $\mathrm{FV}(x) = \{x\}$; $\mathrm{FV}(\lambda x.M) = \mathrm{FV}(M) - \{x\}$ $\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$ **Bound**: 在某λ的 scope 内 **Free**: 不在任何 abstraction 的 scope 内

Combinatory Logic: $\boldsymbol{I}x = x$; $\boldsymbol{K}xy = x$; $\boldsymbol{S}xyz = xz(yz)$ $\boldsymbol{B}xyz = x(yz)$ (comp); $\boldsymbol{C}xyz = xzy$ (flip)

$Txy = yx$ (type-raising) $I = SKK$ (S,K 构成 complete basis)

CCG Rules: **Application**: $X/Y\ Y \Rightarrow X$ (>前向); $Y\ X \setminus Y \Rightarrow X$ (<后向) **Composition**: $X/Y\ Y/Z \Rightarrow X/Z$ ($\boldsymbol{B}_>$) **Type-raising**: $X \Rightarrow T/(T \setminus X)$ ($\boldsymbol{T}_>$) rules 是 universal, language-specific 全在 lexicon

CCG Category 直觉: $S \setminus$ NP: 左边要 NP → 产出 S (intransitive) $(S \setminus \text{NP})/\text{NP}$: 右边要 NP → $S \setminus$ NP (transitive) **Slash 方向**: / 向右找 arg; \ 向左找 arg

LIG 构造策略: **问题**: CFG 无法"计数" ($a^n b^n c^n$中n相等) **LIG**: 用 stack 记录计数信息 **策略 1**: 两端向中间—先生成首尾, 再生成中间 **策略 2**: 左向右—前半部分 push, 后半部分 pop **Example** $a^n b^n c^n d^n$: $S[\sigma] \to aS[f\sigma]d$; $S[\sigma] \to T[\sigma]\ T[f\sigma]$; $T[f\sigma] \to bT[\sigma]c$; $T[] \to \varepsilon$

FOL Translation: ∀配⇒:全称限定条件; ∃配∧: 存在某具体对象; 否则∃配⇒:往往荒谬; ∨配∧:要求满足多个条件.

9. WFST & Lehmann

Transducer Def: $\mathcal{T} = \langle Q, \Sigma, \Omega, \lambda, \rho, \delta \rangle$ Q: states; Σ: input; Ω: output $\lambda : Q \to \mathbb{R}$: initial; $\rho : Q \to \mathbb{R}$: final $\delta : Q \times (\Sigma \cup \varepsilon) \times (\Omega \cup \varepsilon) \times Q \to \mathbb{R}$ **ε-transition**: no input/output consumed

FSA vs FST: **WFSA** (单带): read only, $\text{score}(\pi) = \sum_n \text{score}(\tau_n)$ **WFST** (双带): read input + write output **Unambiguous**: $|\Pi(x,y)| \le 1$ **Ambiguous**: $|\Pi(x,y)| > 1 \longrightarrow$ need semiring

Path Score: $\text{score}(\pi) = \lambda(q_{\text{start}}) + \sum_{n=1}^{|\pi|} \text{score}(\tau_n) + \rho(q_{\text{end}})$ $\quad p(y|x) = \frac{1}{Z} \sum_{\pi \in \Pi(x,y)} \exp(\text{score}(\pi))$ $\quad Z = \sum_{y' \in \Omega^*} \sum_{\pi'} \exp(\text{score}(\pi'))$ (infinite!)

Matrix Mult View: $C = A \otimes B$: $C_{ij} = \bigoplus_k (A_{ik} \otimes B_{kj})$ **Tropical**: $C_{ij} = \min_k (A_{ik} + B_{kj})$ **Inside**: $C_{ij} = \sum_k (A_{ik} \times B_{kj})$ Naive W^N: $O(N^4) \longrightarrow$ Lehmann fixes to $O(N^3)$

Floyd-Warshall: **Key**: allow 中间 node k incrementally $\mathrm{dist}_k[i][j] = \min(\mathrm{dist}_{k-1}[i][j], \mathrm{dist}_{k-1}[i][k] + \mathrm{dist}_{k-1}[k][j])$ Runtime: $O(N^3)$ **FW 是 Lehmann 在 Tropical 的特例** $(a^* = 0)$

Lehmann algo: **Generalized FW** for any closed semiring:

$$W_{ij}^{(k)} = W_{ij}^{(k-1)} \oplus W_{ik}^{(k-1)} \otimes \left(W_{kk}^{(k-1)}\right)^* \otimes W_{kj}^{(k-1)}$$

定义: $\boldsymbol{R}_{ik}^{(j)}$=从$q_i$到$q_k$, 仅经过$\{q_1, ..., q_j\}$的 paths 的 semiring-sum **直觉**: 经过$\{1, ..., j\}$的 paths = 不经j $\oplus$ 经j 后者分解: $i \to j$(不经j) + j上 cycles + $j \to k$(不经j) Runtime: $O(|Q|^3)$

Pathsum & Z: $Z(\mathcal{T}) = \bigoplus_{i,k \in Q} \lambda(q_i) \otimes \boldsymbol{R}_{ik} \otimes \rho(q_k)$ $Z = \alpha^\top \left(\bigoplus_{\omega \in \Sigma^*} W^{(\omega)}\right)^* \beta$ **Why Lehmann?** Direct sum over infinite paths impossible

Composition: $\mathcal{T}(x,y) = \bigoplus_{z \in \Omega^*} \mathcal{T}_1(x,z) \otimes \mathcal{T}_2(z,y)$ **Transliteration**: 3 transducers cascade $\mathcal{T}_x \circ \mathcal{T}_\theta \circ \mathcal{T}_y$

Acyclic WFSA Backward: **前提**: DAG 可做 topological sort

1. 按 reverse topo order 遍历 nodes $q_M, ..., q_1$
2. $\beta[q_m] \leftarrow \rho(q_m) \oplus \bigoplus_{(q_m, a, w, q') \in \delta} w \otimes \beta[q']$
3. return $\bigoplus_{q \in I} \lambda(q) \otimes \beta[q]$

Complexity: $O(|Q| + |\delta|)$ (linear!)

10. Transformers & MT

Seq2Seq: $z = \text{encoder}(x), y|x \sim \text{decoder}(z)$ $p(y|x) = \prod_{t=1}^{T} p(y_t | x, y_1, ..., y_{t-1})$ **Information Bottleneck**: z fixed-length ⟶ Attention 解决

Attention: $\alpha^T V = \sum_i \alpha_i v_i^T$ (soft retrieval) $\alpha_i = \text{softmax}(\text{score}(q, k_i))$ $K = V = H^{(e)}$, $q_t = h_t^{(d)}$, $c = \alpha^T V$

Self-Attention: $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q$, $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K$, $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$ $\text{SelfAtt} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V}$ $\sqrt{\boldsymbol{d_k}}$: 防止 dot product 过大导致 softmax 饱和 **Complexity**: $O(nd^2 + dn^2)$

Positional Encoding: $\boldsymbol{P}_{p,2i} = \sin(p/10000^{2i/d})$ $\boldsymbol{P}_{p,2i+1} = \cos(p/10000^{2i/d})$ motiv: Transformer 无 recurrence, 无法区分位置

Encoder-Decoder 架构: **Encoder**: $+\boldsymbol{P}$ → MHSA → + → LN → MLP → + → LN **Decoder**: +masked self-attn + cross-attn **Masked**: 只 attend 到左边 positions (causal) **Cross-attn**: Q来自 decoder, K, V来自 encoder **Residual**: $x + \text{Layer}(x)$ 缓解 vanishing gradient

Decoding Strategies: $y^* = \arg\max_{y \in \mathcal{Y}} \text{score}(x, y)$ W/o assumptions: $O(|\Sigma|^{n_{\max}})$ paths **Greedy**: 每步 arg max (次优, 快) **Beam**: 保持k-best candidates **Nucleus/Top-p**: 从累积 prob≥ p的 tokens 中 sample **Temperature**: $T < 1$ sharper; $T > 1$ uniform **Eval**: BLEU (n-gram overlap), METEOR

MT Pipeline:

1. **Tokenize**: subword (BPE/WordPiece)
2. **Embed**: token⟶vector + positional
3. **Encode**: Transformer encoder
4. **Decode**: autoregressive, $p(y_n | y_{<n}, \boldsymbol{z})$
5. **Search**: beam/nucleus sampling

Train: MLE, $-\sum \log p(y_n | y_{<n}, \boldsymbol{x})$

11. Modeling Choices

Prob vs Non-Prob: **Prob**: leverage prob theory, needs assumptions CRF, RNN, N-gram models **Non-Prob**: interpretable, uncertainty unclear Perceptron, SVM, CFG rules

Disc vs Generative: **Discriminative**: model boundary $p(y|x)$ **Generative**: model own dist $p(x, y)$

Local vs Global Norm: **Local**: efficient train, biased predictions **Global**: needs Z, unbiased Independence assumptions control complexity

Regularization: **LogLoss**: $\ell(y, y') = \log\left(1 + e^{-y \cdot y'}\right)$ **Exp-Loss**: $\ell(y, y') = e^{-y \cdot y'}$ **L1/L2**: weight penalties (Laplace/Gaussian prior)

Evaluation Metrics: **Prec**: $P_{\text{true}}/P_{\text{all}}$; **Recall**: $P_{\text{true}}/(P_{\text{true}} + N_{\text{false}})$ **Acc**: $(P_{\text{true}} + N_{\text{true}})/N$ **F-score**: $((1 + \beta^2)(\text{prec} \cdot \text{recall}))/(\beta^2 \text{prec} + \text{recall})$

Statistical Tests: $p = 2\min(P(T \ge t | H_0), P(T \le t | H_0))$; Rej if $p < \alpha$ **Power**: $P(\text{reject } H_0 | H_1)$ **Multiple tests**: $P(|\text{FalseRej}| > 0) = 1 - (1 - \alpha)^K$ **Bonferroni**: $\alpha^* = \alpha/K$ **McNemar**: $\chi^2 = \frac{(b-c)^2}{b+c} \sim \chi_1^2$

12. Bias & Fairness

Bias Sources: **Labeling**: reproduce annotator bias **Sample selection**: training fits certain profile **Task definition**: excludes certain groups **Imbalanced test**: loss ignores minorities

Ethical Frameworks: **Consequentialism**: best consequence **Utilitarism**: hedonistic/preference/welfare **Deontology**: rules must be kept **Social Contract**: natural equality **Anti-subordination**: positive discrimination for equality

Quick Ref

Chain: $\frac{d}{dx}[f(g(x))] = f'(g)g'(x)$; Bauer: sum over all paths **Softmax**: $\exp(h_y)/\sum \exp(h_{y'})$; $T \to 0$=argmax **Log-Linear**: $p(y|x) = \exp(\theta \cdot f)/Z$; MLE matches expected features **DP**: distrib 把$O(|T|^N) \to O(N|T|^2)$; 3-gram 则$O(N|T|^3)$ **Fwd/Bwd**: Fwd init BOS+sum last col; Bwd init **1**+single value **Viterbi**: max instead of sum + backpointer **CKY**: $O(N^3|R|)$; CNF; diag first, span 递增 **MTT**: $Z = \det(L)$ in $O(n^3)$; **CLE**: greedy+contract $O(n^2)$ **Lehmann**: $R^{(j)} = R^{(j-1)} \oplus R \otimes R^* \otimes R$; $O(|Q|^3)$ **Kleene**: Inside $\frac{1}{1-a}$; Tropical 0 if $a \ge 0$ **CCG**: $X/Y\ Y \Rightarrow X$ (>); $Y\ X \setminus Y \Rightarrow X$ (<) **β-reduce**: $(\lambda x.M)N \to M[x := N]$; 先$\alpha$-convert 避免捕获 **Self-Attn**: $\text{softmax}\left(QK^T/\sqrt{d}\right)V$; $O(nd^2 + dn^2)$ **Cayley**: 固定 root n^{n-2}; 任意 root n^{n-1}

Abbrev

BOS/EOS: Begin/End of Sentence; **CCG**: Combinatory Categorial Grammar; **CFG**: Context-Free Grammar; **CKY**: Cocke-Kasami-Younger; **CNF**: Chomsky Normal Form; **CRF**: Conditional Random Field; **DP**: Dynamic Programming; **LIG**: Linear Indexed Grammar; **MLE**: Max Likelihood Est; **MST**: Min Spanning Tree; **MTT**: Matrix-Tree Theorem; **POS**: Part-of-Speech; **RNN**: Recurrent Neural Network; **WFST**: Weighted Finite State Transducer;