Autoregressive Ranking: Bridging the Gap Between Dual and Cross Encoders

Benjamin Rozonoyer[1] Chong You[2] Michael Boratko[2] Himanshu Jain[2] Nilesh Gupta[3]
Srinadh Bhojanapalli[2] Andrew McCallum[2] Felix Yu[2]

[1]UMass Amherst [2]Google DeepMind [3]UT Austin

1 Motivation

The dominant IR paradigm is a two-stage pipeline:

- **DEs** enable fast ANN retrieval but compress query–doc interaction into a single dot product, limiting expressivity;
- **CEs** score query–doc pairs jointly but are prohibitively costly for first-stage retrieval.

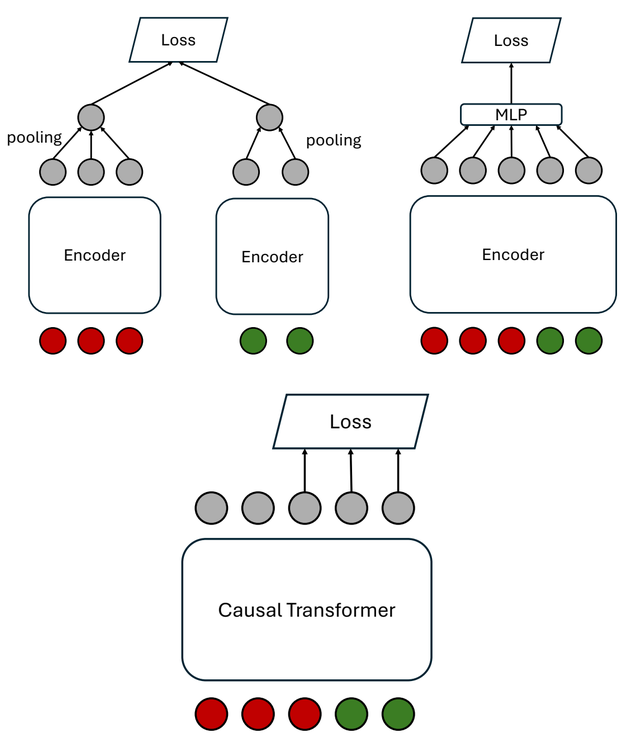


Figure 1: DE vs. CE vs. ARR at a glance.

Autoregressive Ranking (ARR): one LLM generates docIDs token-by-token and ranks them with beam search, aiming for CE-style expressivity without scoring every document individually.

2 Theory: Capacity for Ranking

ARR is strictly more expressive than dual encoders.

DE lower bound

To realize **every** ranking of k documents, a dual encoder needs embedding dimension

$$n = \Omega(k).$$

DE capacity must grow linearly with corpus size.

ARR rank condition

An ARR with **constant** hidden dimension can realize **any** ranking, provided the docID token embedding submatrix E' satisfies $\mathrm{rank}(E') = |\mathcal{V}_{\mathrm{docID}}|$. This condition is *necessary and sufficient* — and strictly weaker than what any dual encoder requires.

Dual Encoder
$n = \Omega(k)$

ARR
$n = O(1)$

3 Rank-Aware Training Loss

Unified formulation, two complementary components.

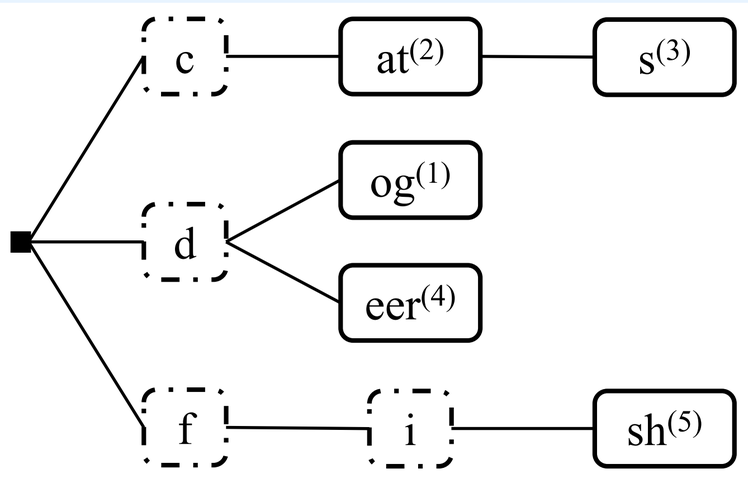
Given (q, d_r, r) *(query, docID, rank)*:

$$\mathcal{L}(q, d_r, r; \theta) = \lambda(r) \sum_{t=1}^{|\mathcal{T}(d_r)|} \mathrm{CE}\big(\mathbf{y}(r,t),\, p_\theta(\mathcal{T}(d_r)_t \mid \mathcal{T}(\bar{q}), \mathcal{T}(d_r)_{<t})\big)$$

Item-level reweighting $\lambda(r)$: fractional $\lambda(r)=1/r^\alpha$ or stepwise $\lambda(r)=(n_q-r+1)/n_q$ weight each document by rank and spread supervision across the full list. $\lambda(r)=\mathbb{I}_{r=1}$ recovers standard NTP.

Token-level marginalization $\mathbf{y}(r,t)$: assign soft target $\mathbf{y}(r,t)$ at each generation step by marginalizing leaf scores $1/r^\beta$ over the docID prefix tree.

`dog` ≻ `cat` ≻ `cats` ≻ `deer` ≻ `fish`



Prefix tree over ranked, tokenized docIDs

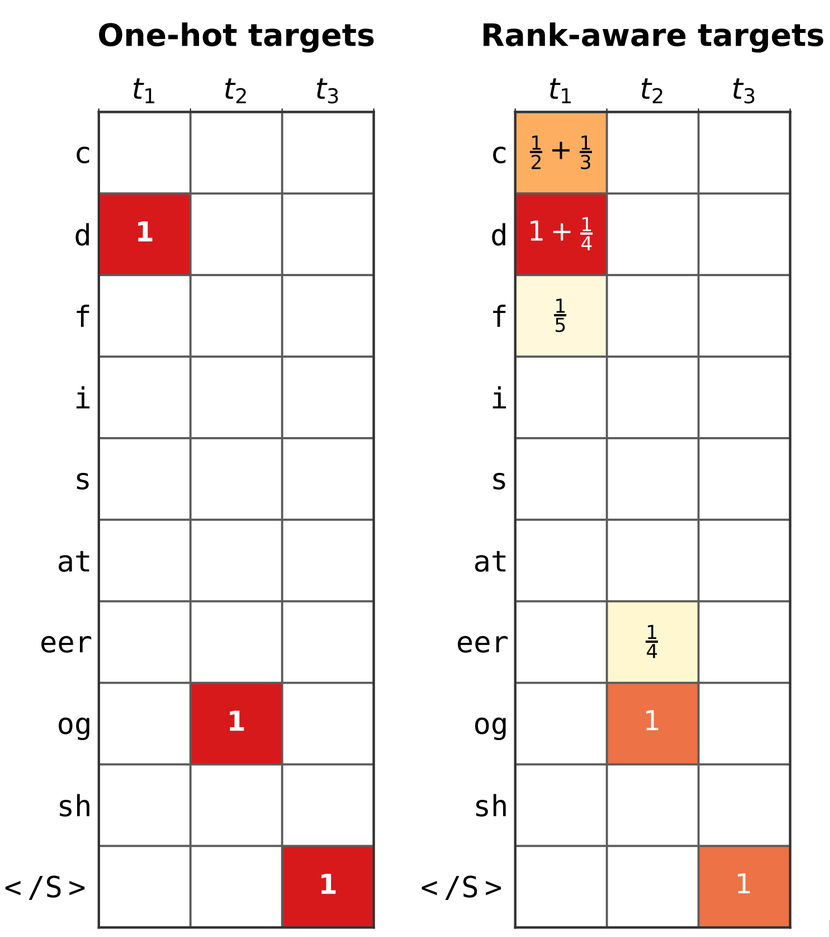


Figure 2: Leaf scores $1/r^\beta$ marginalized into per-step soft targets (b) vs. one-hot targets (a).

4 WordNet Results

Fractional reweighting eliminates violations and lifts recall at all ranks.

Model: Mistral-7B-v0.3-it. CVR = Constraint Violation Rate.

Loss	CVR↓	nDCG↑	R@1↑	R@2↑	R@3↑	R@5↑
NTP	27.7%	94.9	99.96	62.4	55.3	63.4
$\lambda{=}1/r^1$, one-hot	0.0%	99.6	91.1	87.7	88.8	93.9
$\lambda{=}1/r^2$, one-hot	0.0%	99.8	97.7	95.7	95.5	96.6
$\lambda{=}1/r^3$, one-hot	0.0%	99.8	99.2	97.4	96.2	95.5
Stepwise, one-hot	0.0%	99.6	51.6	69.9	82.1	93.3
$\mathbb{I}_{r=1}$, trie $\beta{=}1$	1.5%	96.5	98.1	72.9	63.1	64.3
$\mathbb{I}_{r=1}$, trie $\beta{=}2$	1.4%	96.5	99.3	67.3	57.5	62.5

Table 1. WordNet: $\lambda{=}1/r^2$ is optimal (0% CVR, best R@2–R@5 balance).

5 ESCI Results

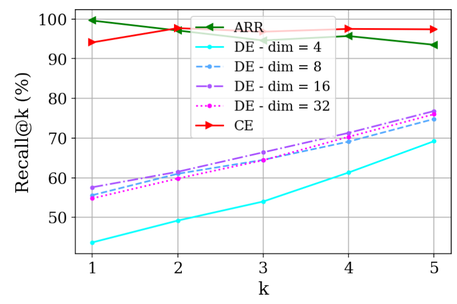
One trie target per query is enough to improve nDCG and recall.

Loss	nDCG↑	R@1↑	R@2↑	R@5↑	R@10↑	R@25↑	R@50↑
NTP	95.2	95.2	52.6	27.6	23.5	38.0	63.0
Trie $\beta{=}1$	97.2	70.0	56.6	45.6	46.3	54.8	69.6
Trie $\beta{=}2$	97.2	70.3	58.1	51.1	48.1	55.0	69.0
Trie $\beta{=}3$	97.1	68.7	56.1	48.0	45.8	52.3	67.6

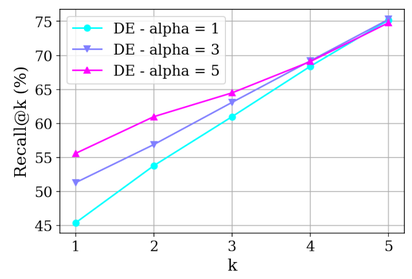
Table 2. ESCI: trie targets improve nDCG (+2.0) and R@5 (+23.5) over NTP.

6 ARR vs. DE/CE (WordNet)

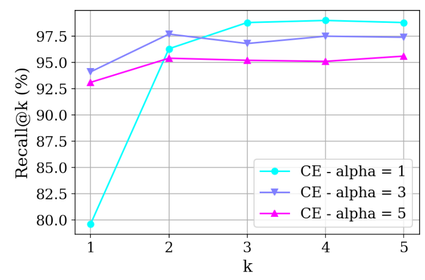
ARR closes the gap to CE; rank-aware λ helps both DE and CE.



(a) ARR and CE saturate recall; DEs plateau.



(b) α helps DEs, but the ceiling remains.



(c) Rank-aware reweighting also lifts CE recall.

Figure 3: ARR matches CE; rank-aware λ improves both DE and CE, but only ARR closes the gap.

ARR, generating multi-token docIDs, achieves CE-level expressivity at *constant* hidden dimension — something no dual encoder can do.

Rank-aware reweighting λ and prefix-tree soft targets $\mathbf{y}$ are two orthogonal, plug-in improvements to standard NTP.

Rank-aware training suppresses invalid decoding and improves recall at larger k. Only ARR reaches CE-level performance.