$G(q)$ behavior: $\overline{IDF}(q) \ll \pi_m$ (common words in query), $G(q) \approx 0$ and $R(q^{mic}, d^{mic})$ doesn't contribute

$\overline{IDF}(q) \gg \pi_m$ (rare queries), $G(q) \approx 1$ and $R(q^{mic}, d^{mic})$ is weighted by w_{mic} (0.12)

The double log structure for term weight: additinal matched terms do not contribute additatively, but provide diminishing returns

B_{cov}: A document match all query terms gets a $1+\gamma_{cov}$ multiplier (×1.25)
match none gets ×1 multiplier, but in that case, E is also 0

$PMI(t,d)$: kind of like $\log\left(\frac{P(t|d)}{P(t|c)}\right)$ $\quad P(t|d) = \frac{tf(t,d)}{|d|_{eff}}$ $\quad P(t|c) = \frac{df(t)}{N}$

B_{spec}: how much information the observation "t is in d" carries beyond what the corpus prior would predict

B_{anc}: if no matched term has $IDF(t) > \pi$ (4.2) then $A=0$ and $B_{anc}=1$

Optimal Multiplier implemented by GPT 5.2 but was not used ($\alpha_{gini}=0$)

$$B_{gini} = 1 + \alpha_{gini} \cdot Conc(q,d)$$

$$Conc(q,d) = clip\left(\frac{H - \frac{1}{|M|}}{1 - \frac{1}{|M|}}, 0, 1\right)$$

Conc high → If document's evidence is concentrated on a few terms, (topical focus?)
Conc low if evidence spreads evenly

$$H = \frac{\sum_{t \in M} (w(t) \cdot tf(t,d))^2}{\left(\sum_{t \in M} (w(t) \cdot tf(t,d))\right)^2}$$

Tokenization Channels:

base: Lucene tokenizer

prefix: neural → neura, network → netwo
(only keep first L_p characters $L_p=5$)

Bigram: [neural, network, model] → [neural·network, network·model]

Micro: [neural] → [neu, eur, ura, ral]
(all overlapping L_m-grams, $L_m=3$)