$P_c(t)$ ~~serves~~ serves as collection prior

$$P_c(t) = (1-\overset{0.03}{\gamma}) \cdot P_{mix}(t) + \gamma \cdot \frac{1}{|V|}$$

$P_{mix}(t)$ is similar to linear smoothing $\longrightarrow P_{mix}(t) = (1-\overset{0.1}{\alpha_{df}}) \cdot P_\tau(t) + \alpha_{df} P_{df}(t)$

$P_\tau(t)$ is like the collection LM, and it looks like a softmax function, τ is like temperature, controlling how peaked the background LM.

$$P_\tau(t) = \frac{P(t|C)^\tau}{\sum_{t' \in V} P(t'|C)^\tau}$$

$P_{df}(t)$ is new, it's like a document-presence LM

$$P_{df}(t) = \frac{df(t)}{N}$$

S_{base} in $\tilde{S}$ would be negative if

$$tf(t,d)^{\beta(t)} \leq |d| \cdot P_c(t)$$

meaning the term is present in the document, but at a lower rate than the collection model would have predicted for a document of that length

$\frac{P_{df}(t)}{P_c(t)}$ if a term appears in many documents but has low total frequency, it is more informative.
if a term has low document frequency but has a lot of term frequency, it is concentrated in a few documents

$g(t) = 1 + \lambda_{edr} \cdot \text{clip}\left(\log\frac{P_{df}(t)}{P_c(t)}, -C_{edr}, C_{edr}\right) \longrightarrow$ this modifies each query term's weight

$m(t,d)$ is always negtive, it's a penalty for if the query term is not in the document

$AND(q,d)$ rewards the document if it has more distinct qury terms, tanh is almost like a binary func

$LP(d)$ is the length penalty for documents longer than corpus average on a log scale