Bayesian BM25: A Probabilistic Framework for Hybrid Text and Vector Search

Jaepil Jeong
Cognica, Inc.
Email: jaepil@cognica.io
Date: January 23, 2026

"The theory of probabilities is at bottom nothing but common sense reduced to calculus."
— Pierre-Simon Laplace

Abstract

This paper presents Bayesian BM25, a novel probabilistic framework that transforms traditional BM25 relevance scores into calibrated probability estimates, enabling principled integration of lexical and semantic search signals. We demonstrate that standard BM25 scores, while effective for ranking, suffer from fundamental interpretability limitations: unbounded range, query-dependent magnitudes, and incompatibility with other ranking signals. We address these limitations by applying Bayesian inference with a sigmoid likelihood model and composite prior design incorporating term frequency and document length signals. We introduce a corpus-level base rate prior that decomposes the posterior into three additive log-odds terms, reducing expected calibration error by 68--77% without requiring relevance labels. We prove that our transformation preserves the monotonicity properties essential for ranking while producing well-calibrated probability outputs in $[0, 1]$. We extend this framework to hybrid search, deriving probabilistic score combination formulas for Boolean query operations under independence assumptions. We show that log-space computation ensures numerical stability for extreme probabilities. Furthermore, we analyze the compatibility of Bayesian BM25 with WAND (Weak AND) and BMW (Block-Max WAND) optimization algorithms, proving that the transformation preserves safe document pruning through modified upper bounds that account for document-dependent priors. Experimental results demonstrate that Bayesian BM25 achieves comparable ranking quality to standard BM25 while enabling principled multi-signal fusion without ad-hoc normalization or weighting schemes.

1. Introduction and Motivation

1.1 The BM25 Score Interpretation Problem

The BM25 (Best Matching 25) scoring function, introduced by Robertson and Zaragoza (2009), has become the de facto standard for lexical relevance scoring in information retrieval systems. Despite its empirical success, BM25 scores present fundamental challenges for modern search applications that require combining multiple ranking signals.

Definition 1.1.1 (BM25 Score). For a document D and query $Q = \{t_1, t_2, \ldots, t_m\}$, the BM25 score is defined as:

$$\text{BM25}(D, Q) = \sum_{t \in Q} \text{IDF}(t) \cdot \frac{f(t, D) \cdot (k_1 + 1)}{f(t, D) + k_1 \cdot B(D)} \tag{1}$$

where:

- $f(t, D)$ is the term frequency of t in document D
- $\text{IDF}(t)$ is the inverse document frequency of term t
- $B(D) = 1 - b + b \cdot \frac{|D|}{\text{avgdl}}$ is the length normalization factor
- k_1 and b are tuning parameters (typically $k_1 = 1.2$ and $b = 0.75$)

Problem 1.1.2 (Score Interpretability). BM25 scores suffer from the following interpretability limitations:

1. **Unbounded Range**: $\text{BM25}(D, Q) \in [0, +\infty)$, making absolute interpretation impossible
2. **Query Dependence**: Score magnitudes vary with query length and term specificity
3. **Corpus Dependence**: IDF values depend on collection statistics, preventing cross-corpus comparison
4. **Signal Incompatibility**: Direct combination with bounded signals (e.g., vector similarity in $[0, 1]$) leads to dominated or unbalanced contributions

1.2 The Multi-Signal Fusion Challenge

Modern search systems require combining heterogeneous ranking signals with different scales and distributions.

Definition 1.2.1 (Multi-Signal Ranking Function). A multi-signal ranking function combines n signals:

$$\text{Score}(D, Q) = f(s_1(D, Q), s_2(D, Q), \ldots, s_n(D, Q)) \tag{2}$$

where each $s_i : \mathcal{D} \times \mathcal{Q} \to \mathbb{R}$ is a scoring function with potentially different range and distribution.

Theorem 1.2.2 (Signal Dominance in Naive Combination). For additive combination of signals with different scales:

$$\text{Score}_{naive} = \sum_{i=1}^{n} s_i \tag{3}$$

the signal with the largest expected magnitude will dominate the ranking.

Proof. Let $\mu_i = \mathbb{E}[s_i]$ and $\sigma_i = \text{Std}[s_i]$. For two signals with $\mu_1 \gg \mu_2$, the probability that signal 2 affects the ranking is:

$$P(s_2 > s_1) \leq P\left(\frac{s_1 - \mu_1}{\sigma_1} < -\frac{\mu_1 - \mu_2}{\sigma_1}\right) \approx 0 \tag{4}$$

when $\mu_1 - \mu_2 \gg \sigma_1$.

1.3 Reciprocal Rank Fusion Limitations

Reciprocal Rank Fusion (RRF), proposed by Cormack et al. (2009), is a common approach to combining ranked lists.

Definition 1.3.1 (Reciprocal Rank Fusion). For n ranked lists $R_1, \ldots, R_n$, the RRF score is:

$$\mathrm{RRF}(D) = \sum_{i=1}^{n} \frac{1}{k + \mathrm{rank}_i(D)} \tag{5}$$

where k is a constant (typically $k = 60$) and $\mathrm{rank}_i(D)$ is the rank of document D in list R_i.

Theorem 1.3.2 (RRF Limitations). RRF suffers from the following theoretical limitations:

1. **Information Loss**: Score magnitudes are discarded, losing confidence information
2. **Rank Sensitivity**: Small perturbations in scores can cause large rank changes
3. **Non-Commutativity with Filtering**: $\mathrm{RRF}(\mathrm{Filter}(L_1), \mathrm{Filter}(L_2)) \neq \mathrm{Filter}(\mathrm{RRF}(L_1, L_2))$
4. **Arbitrary Constant**: The k parameter lacks theoretical justification
5. **Missing Document Handling**: Undefined behavior for documents appearing in only some lists

Proof of (3). Consider $L_1 = [A : 10, B : 5, C : 3]$ and $L_2 = [B : 8, A : 4, C : 1]$. With filter threshold > 6:

- Left side: $\mathrm{RRF}([A : 10], [B : 8]) = \{A : \frac{1}{61}, B : \frac{1}{61}\}$
- Right side: $\mathrm{Filter}(\mathrm{RRF}(L_1, L_2)) = \mathrm{Filter}(\{A : \frac{1}{61} + \frac{1}{62}, B : \frac{1}{62} + \frac{1}{61}, C : \frac{1}{63} + \frac{1}{63}\})$

The results differ because filtering after fusion considers documents that wouldn't pass individual filters.

1.4 Our Contribution

We propose Bayesian BM25, a probabilistic framework that:

1. Transforms BM25 scores into calibrated probabilities in $[0, 1]$
2. Preserves monotonicity properties essential for ranking
3. Enables principled probabilistic combination of multiple signals
4. Maintains compatibility with WAND/BMW optimization algorithms through modified upper bounds
5. Supports online parameter learning for domain adaptation
6. Introduces a corpus-level base rate prior for absolute probability calibration without relevance labels

2. Mathematical Preliminaries

2.1 Probability Calibration

Definition 2.1.1 (Calibrated Probability). A scoring function $s : \mathcal{D} \times \mathcal{Q} \to [0, 1]$ is calibrated if:

$$P(\text{relevant} \mid s(D, Q) = p) = p \tag{6}$$

for all $p \in [0, 1]$.

Definition 2.1.2 (Sigmoid Function). The sigmoid function $\sigma : \mathbb{R} \to (0, 1)$ is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

Lemma 2.1.3 (Sigmoid Properties). The sigmoid function satisfies:

1. $\sigma(-x) = 1 - \sigma(x)$ (symmetry)
2. $\frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x))$ (derivative)
3. $\lim_{x \to \infty} \sigma(x) = 1$ and $\lim_{x \to -\infty} \sigma(x) = 0$ (bounds)

Proof. Property (1):

$$\sigma(-x) = \frac{1}{1 + e^{x}} = \frac{e^{-x}}{e^{-x} + 1} = 1 - \frac{1}{1 + e^{-x}} = 1 - \sigma(x) \tag{8}$$

2.2 Bayesian Inference Framework

Definition 2.2.1 (Bayes' Theorem). For hypothesis H and evidence E:

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)} \tag{9}$$

Definition 2.2.2 (Binary Classification Posterior). For relevance $R \in \{0, 1\}$ and score s:

$$P(R = 1 \mid s) = \frac{P(s \mid R = 1) \cdot P(R = 1)}{P(s \mid R = 1) \cdot P(R = 1) + P(s \mid R = 0) \cdot P(R = 0)} \tag{10}$$

3. BM25 Theoretical Foundation

3.1 IDF Computation

Definition 3.1.1 (Robertson-Sparck Jones IDF). The inverse document frequency with smoothing is:

$$\mathrm{IDF}(t) = \ln\left(\frac{N - \mathrm{df}(t) + 0.5}{\mathrm{df}(t) + 0.5} + 1\right) \tag{11}$$

where N is the total document count and $\mathrm{df}(t)$ is the document frequency of term t.

Theorem 3.1.2 (IDF Non-Negativity). For any term t with $\mathrm{df}(t) \leq N$:

$$\mathrm{IDF}(t) \geq 0 \tag{12}$$

Proof. The argument of the logarithm is:

$$\frac{N - \mathrm{df}(t) + 0.5}{\mathrm{df}(t) + 0.5} + 1 \geq 1 \tag{13}$$

since $N - \mathrm{df}(t) \geq 0$. Therefore, $\mathrm{IDF}(t) = \ln(\cdot) \geq \ln(1) = 0$.

Theorem 3.1.3 (IDF Monotonicity). IDF is monotonically decreasing in document frequency:

$$\mathrm{df}(t_1) < \mathrm{df}(t_2) \implies \mathrm{IDF}(t_1) > \mathrm{IDF}(t_2) \tag{14}$$

Proof. Let $f(x) = \ln\left(\frac{N - x + 0.5}{x + 0.5} + 1\right)$. Taking the derivative:

$$f'(x) = \frac{-(N + 1)}{(x + 0.5)(N - x + 0.5) + (x + 0.5)^2} < 0 \tag{15}$$

for $0 \leq x \leq N$, proving monotonicity.

3.2 Monotonicity-Preserving BM25 Formulation

Definition 3.2.1 (Rewritten BM25 Scoring). We express BM25 in a numerically stable form:

$$\text{score}(f, n) = w - \frac{w}{1 + f \cdot \text{inv\_norm}} \tag{16}$$

where:

- $w = \text{boost} \cdot \text{IDF}$ (weight)
- $\text{inv\_norm} = \frac{1}{k_1 \cdot ((1-b) + b \cdot \frac{n}{\text{avgdl}})}$ (inverse normalization)
- f is term frequency
- n is document length

Theorem 3.2.2 (BM25 Monotonicity). The BM25 score satisfies:

1. **Monotonic in term frequency**: $f_1 > f_2 \implies \text{score}(f_1, n) > \text{score}(f_2, n)$
2. **Anti-monotonic in document length**: $n_1 > n_2 \implies \text{score}(f, n_1) < \text{score}(f, n_2)$

Proof of (1). Let $g(f) = w - \frac{w}{1+f\cdot c}$ where $c = \text{inv\_norm} > 0$. Then:

$$g'(f) = \frac{w \cdot c}{(1 + f \cdot c)^2} > 0 \tag{17}$$

proving monotonicity in f.

Proof of (2). Let $h(n) = w - \frac{w}{1+f/(k_1\cdot((1-b)+b\cdot n/\text{avgdl}))}$. As n increases, the denominator of inv_norm increases, making inv_norm smaller, which decreases the overall score.

Theorem 3.2.3 (BM25 Upper Bound). For any term with weight w:

$$\sup_{f,n} \text{score}(f, n) = w = \text{boost} \cdot \text{IDF} \tag{18}$$

Proof. As $f \to \infty$:

$$\lim_{f\to\infty} \left(w - \frac{w}{1 + f \cdot c} \right) = w - 0 = w \tag{19}$$

---

4. Bayesian BM25 Framework

4.1 Probabilistic Model

Definition 4.1.1 (Likelihood Model). We model the probability of observing score s given relevance R using a parametric sigmoid:

$$P(s \mid R = 1) = \sigma(\alpha \cdot (s - \beta)) = \frac{1}{1 + e^{-\alpha(s-\beta)}} \tag{20}$$

where:

- $\alpha > 0$ controls the sigmoid steepness (score sensitivity)
- β controls the sigmoid midpoint (decision boundary)

Definition 4.1.2 (Symmetric Likelihood Assumption). We assume:

$$P(s \mid R=0) = 1 - P(s \mid R=1) = \sigma(-\alpha \cdot (s-\beta)) \tag{21}$$

Theorem 4.1.3 (Posterior Probability Formula). Under the symmetric likelihood assumption, the posterior probability of relevance is:

$$P(R=1 \mid s) = \frac{L \cdot p}{L \cdot p + (1-L) \cdot (1-p)} \tag{22}$$

where $L = \sigma(\alpha(s-\beta))$ is the likelihood and $p = P(R=1)$ is the prior.

Proof. Applying Bayes' theorem with the symmetric likelihood:

$$P(R=1 \mid s) = \frac{P(s \mid R=1) \cdot P(R=1)}{P(s \mid R=1) \cdot P(R=1) + P(s \mid R=0) \cdot P(R=0)} \tag{23}$$

$$= \frac{L \cdot p}{L \cdot p + (1-L) \cdot (1-p)} \tag{24}$$

4.2 Composite Prior Design

Definition 4.2.1 (Term Frequency Prior). The prior probability based on term frequency is:

$$P_{\text{tf}}(f) = 0.2 + 0.7 \cdot \min\left(1, \frac{f}{10}\right) \tag{25}$$

Definition 4.2.2 (Field Norm Prior). The prior probability based on document length normalization is:

$$P_{\text{norm}}(n) = 0.3 + 0.6 \cdot (1 - \min(1, |n-0.5| \cdot 2)) \tag{26}$$

Definition 4.2.3 (Composite Prior). The combined prior probability is:

$$P_{\text{prior}}(f,n) = \text{clamp}\left(0.7 \cdot P_{\text{tf}}(f) + 0.3 \cdot P_{\text{norm}}(n), 0.1, 0.9\right) \tag{27}$$

Theorem 4.2.4 (Prior Bounds). For all valid inputs:

$$0.1 \leq P_{\text{prior}}(f,n) \leq 0.9 \tag{28}$$

Proof. The clamp function explicitly enforces these bounds, preventing extreme priors from dominating the posterior.

Remark 4.2.5 (Empirical Bayes Justification). The composite prior $P_{\text{prior}}(f, \hat{n})$ uses the same features — term frequency f and encoded field norm $\hat{n}$ — that contribute to the BM25 score $s(f, \hat{n})$. In a strictly classical Bayesian framework, the prior should be independent of the observed evidence. Our design constitutes an **empirical Bayes** approach (Robbins, 1956; Efron, 2010), in which the prior is informed by observed data features through a different functional form than the likelihood.

This is standard practice in hierarchical modeling: the James-Stein estimator, empirical Bayes confidence intervals, and parametric empirical Bayes methods all estimate prior hyperparameters from data. The key distinction is that BM25 combines f and $\hat{n}$ through a specific nonlinear function parametrized by (k_1, b) to produce a **relevance score**, while the prior combines them through a different function to encode a **document quality signal** — the meta-level observation that documents with moderate length and non-trivial term frequency are more likely to be genuinely relevant, as opposed to spam, boilerplate, or incidental matches.

For applications requiring strict prior-evidence independence, the prior can be constructed from features that do not enter the BM25 score: document recency, source authority (PageRank), content type, or other static quality signals. Definition 4.2.3 provides a self-contained prior that requires no external metadata, at the cost of the empirical Bayes approximation.

4.3 Monotonicity Preservation

Theorem 4.3.1 (Bayesian BM25 Monotonicity). The Bayesian BM25 transformation preserves the monotonicity of BM25 scores:

$$s_1 > s_2 \implies P(R=1 \mid s_1) > P(R=1 \mid s_2) \tag{29}$$

for fixed prior p and $\alpha > 0$.

Proof. Let $g(s) = P(R=1 \mid s)$ with fixed prior p. We show $g'(s) > 0$.

Let $L(s) = \sigma(\alpha(s-\beta))$. Then $L'(s) = \alpha \cdot L(s) \cdot (1 - L(s)) > 0$.

$$g(s) = \frac{L \cdot p}{L \cdot p + (1-L) \cdot (1-p)} \tag{30}$$

Let $A = L \cdot p$ and $B = (1-L) \cdot (1-p)$. Then $g = \frac{A}{A+B}$.

$$g' = \frac{A'(A+B) - A(A'-B')}{(A+B)^2} = \frac{A' \cdot B + A \cdot B'}{(A+B)^2} \tag{31}$$

Since $A' = L' \cdot p > 0$ and $B' = -L' \cdot (1-p) < 0$:

$$g' = \frac{L' \cdot p \cdot (1-L)(1-p) + L \cdot p \cdot L'(1-p)}{(A+B)^2} = \frac{L' \cdot p \cdot (1-p)}{(A+B)^2} > 0 \tag{32}$$

Remark 4.3.2 (Scope of Monotonicity). Theorem 4.3.1 establishes that the posterior is monotone in s **for fixed prior** p. When the composite prior $P_{\text{prior}}(f, \hat{n})$ varies across documents (Definition 4.2.3), the posterior is a function of both s and p, and the overall ranking may differ from the BM25 ranking. This has implications for WAND pruning safety — see Theorem 6.1.2 for the corrected upper bound that accounts for the prior's variability.

4.4 Base Rate Prior

The auto-estimation heuristic of Section 8.3 sets $\beta = \text{median}(\text{scores})$, placing the sigmoid midpoint at the corpus median. Because most documents are not relevant to a typical query, this assigns approximately 50% relevance probability to scores that are almost never relevant, producing systematic overconfidence. The document-dependent prior (Definition 4.2.3) cannot correct this because it operates per-document and does not encode the corpus-level prevalence of relevance.

We introduce a corpus-level base rate that captures the global prior probability that a randomly selected document is relevant to a randomly selected query.

Definition 4.4.1 (Base Rate Prior). The base rate prior $b_r \in (0,1)$ is the corpus-level prior probability that a document is relevant to a query, independent of any document-specific features. For a corpus of N documents and query distribution $\mathcal{Q}$:

$$b_r = \mathbb{E}_{q \sim \mathcal{Q}} \left[\frac{|\{d : d \text{ is relevant to } q\}|}{N} \right] \tag{33}$$

Theorem 4.4.2 (Three-Term Posterior Decomposition). Let $L = \sigma(\alpha(s - \beta))$ be the sigmoid likelihood (Definition 4.1.1), b_r be the base rate prior (Definition 4.4.1), and $p = P_{\text{prior}}(f, \hat{n})$ be the composite prior (Definition 4.2.3). The full posterior decomposes as three additive terms in log-odds space:

$$\operatorname{logit}(P(R = 1 \mid s, f, \hat{n})) = \operatorname{logit}(L) + \operatorname{logit}(b_r) + \operatorname{logit}(p) \tag{34}$$

or equivalently:

$$P(R = 1 \mid s, f, \hat{n}) = \sigma(\operatorname{logit}(L) + \operatorname{logit}(b_r) + \operatorname{logit}(p)) \tag{35}$$

Proof. We apply two successive Bayes updates. The first update combines the likelihood L with the composite prior p via Theorem 4.1.3:

$$P_1 = \frac{L \cdot p}{L \cdot p + (1 - L)(1 - p)} \tag{36}$$

Converting to log-odds using $\operatorname{logit}(x) = \ln \frac{x}{1-x}$:

$$\operatorname{logit}(P_1) = \ln \frac{L \cdot p}{(1 - L)(1 - p)} = \ln \frac{L}{1 - L} + \ln \frac{p}{1 - p} = \operatorname{logit}(L) + \operatorname{logit}(p) \tag{37}$$

The second Bayes update treats P_1 as a new likelihood and b_r as the prior:

$$P_2 = \frac{P_1 \cdot b_r}{P_1 \cdot b_r + (1 - P_1)(1 - b_r)} \tag{38}$$

Again converting to log-odds:

$$\operatorname{logit}(P_2) = \operatorname{logit}(P_1) + \operatorname{logit}(b_r) = \operatorname{logit}(L) + \operatorname{logit}(p) + \operatorname{logit}(b_r) \tag{39}$$

Since $\sigma(\operatorname{logit}(x)) = x$, we recover $P_2 = \sigma(\operatorname{logit}(L) + \operatorname{logit}(b_r) + \operatorname{logit}(p))$. □

Corollary 4.4.3 (Uniform Base Rate Neutrality). When $b_r = 0.5$:

$$\operatorname{logit}(0.5) = \ln \frac{0.5}{0.5} = 0 \tag{40}$$

so the three-term decomposition reduces to the two-term form $\operatorname{logit}(P) = \operatorname{logit}(L) + \operatorname{logit}(p)$, recovering Theorem 4.1.3.

Theorem 4.4.4 (Monotonicity with Base Rate). For fixed base rate $b_r \in (0, 1)$ and fixed prior $p \in (0, 1)$, the posterior $P(R = 1 \mid s)$ is strictly monotonically increasing in the BM25 score s.

Proof. The posterior is $P = \sigma(\operatorname{logit}(L) + c)$ where $c = \operatorname{logit}(b_r) + \operatorname{logit}(p)$ is a constant. Since $L(s) = \sigma(\alpha(s - \beta))$ is strictly increasing in s (as $\alpha > 0$), $\operatorname{logit}(L(s)) = \alpha(s - \beta)$ is strictly increasing in s. Adding the constant c preserves strict monotonicity, and σ is strictly increasing. The composition of strictly increasing functions is strictly increasing. □

Remark 4.4.5 (Efficient Computation). Rather than evaluating the full three-term log-odds formula (which requires two $\ln$ and one $\exp$ per term), the implementation uses two successive Bayes updates in probability space:

1. Compute $P_1 = \frac{L \cdot p}{L \cdot p + (1 - L)(1 - p)}$ (existing two-term posterior)
2. Compute $P_2 = \frac{P_1 \cdot b_r}{P_1 \cdot b_r + (1 - P_1)(1 - b_r)}$ (base rate update)

Step 2 requires only 2 multiplications, 1 subtraction, and 1 division -- approximately 13% additional overhead beyond the two-term posterior. This is mathematically equivalent to the log-odds formulation (Theorem 4.4.2) but avoids the transcendental function evaluations.

Algorithm 4.4.7 (Base Rate Estimation). In the absence of relevance labels, we estimate b_r from the corpus score distribution using pseudo-queries:

```
Input: Corpus C = [d_1, ..., d_N], BM25 index I
Output: Estimated base rate b_r

m = min(N, 50)
indices = sample_uniform(1..N, m)

for i = 1 to m:
    q_i = first_5_tokens(C[indices[i]])
    S_i = {s(d, q_i) : s > 0 for all d in C}
    t_i = percentile(S_i, 95)
    r_i = |{s in S_i : s >= t_i}| / N

b_r = clamp(mean(r_1, ..., r_m), 1e-6, 0.5)

return b_r
```

The 95th percentile threshold selects documents with unusually high scores relative to the query -- a proxy for the fraction of "highly relevant" documents. The upper clamp at 0.5 ensures $\text{logit}(b_r) \leq 0$, so the base rate can only shift the posterior downward (toward lower relevance probability), correcting the overconfidence introduced by the median-based β estimate.

Remark 4.4.8 (Relationship to C1--C3 Conditions). The base rate prior complements the consistency conditions of Remark 8.3.0. Conditions C1--C3 require labeled data to correctly separate the likelihood from prior information during training. The base rate prior provides an orthogonal, unsupervised calibration mechanism: even without relevance labels, estimating b_r from the corpus score distribution reduces ECE by 68--77% (Section 11.3). When labeled data is available, the base rate prior can be combined with batch fitting (Algorithm 8.3.1) for further improvement, or disabled ($b_r = 0.5$, Corollary 4.4.3) if the supervised calibration is sufficient.

5. Hybrid Search Score Combination

5.1 Probabilistic Conjunction (AND)

Theorem 5.1.1 (Probabilistic AND). For independent relevance events with probabilities $p_1, p_2, \ldots, p_n$:

$$P(R_1 \wedge R_2 \wedge \cdots \wedge R_n) = \prod_{i=1}^{n} p_i \tag{41}$$

Definition 5.1.2 (Log-Space Conjunction). For numerical stability, we compute:

$$P(\text{AND}) = \exp\left(\sum_{i=1}^{n} \ln(p_i)\right) \tag{42}$$

Theorem 5.1.2 (Conjunction Bounds). For $p_i \in (0, 1)$:

$$0 < P(\text{AND}) < \min_i p_i \tag{43}$$

Proof. Since each $\ln(p_i) < 0$ for $p_i < 1$, the sum is negative, and $\exp$ of a negative number is less than 1. The product is strictly less than any individual factor.

5.2 Probabilistic Disjunction (OR)

Theorem 5.2.1 (Probabilistic OR). For independent relevance events:

$$P(R_1 \vee R_2 \vee \cdots \vee R_n) = 1 - \prod_{i=1}^{n}(1 - p_i) \tag{44}$$

Definition 5.2.2 (Log-Space Disjunction). For numerical stability:

$$P(\text{OR}) = 1 - \exp\left(\sum_{i=1}^{n} \ln(1 - p_i)\right) \tag{45}$$

Theorem 5.2.2 (Disjunction Bounds). For $p_i \in (0, 1)$:

$$\max_i p_i < P(\text{OR}) < 1 \tag{46}$$

Proof. Since $1 - p_i < 1$ for $p_i > 0$, the product $\prod(1 - p_i) > 0$, so $P(\text{OR}) < 1$. The disjunction probability exceeds any individual probability because additional positive probabilities can only increase the result.

5.3 Numerical Stability Analysis

Theorem 5.3.1 (Log-Space Stability). Computing probabilities in log-space avoids underflow for extreme probabilities:

$$\ln(p_1 \cdot p_2 \cdots p_n) = \sum_{i=1}^{n} \ln(p_i) \tag{47}$$

remains representable when direct multiplication would underflow.

Proof. For IEEE 754 double precision, the minimum positive value is approximately 2.2×10^{-308}. Computing $\prod_{i=1}^{100} 0.01$ directly would yield 10^{-200}, which underflows. In log-space: $\sum_{i=1}^{100} \ln(0.01) = -460.5$, which is representable, and $\exp(-460.5)$ can be clamped or handled gracefully.

Definition 5.3.2 (Probability Clamping). To ensure numerical stability:

$$p_{\text{safe}} = \text{clamp}(p, \epsilon, 1 - \epsilon) \tag{48}$$

where $\epsilon = 10^{-10}$ is a small constant.

6. WAND and BMW Optimization

6.1 WAND Algorithm

Definition 6.1.1 (WAND Condition). The WAND (Weak AND) algorithm skips documents when:

$$\sum_{i=0}^{\text{pivot}} \text{upper\_bound}_i < \theta \tag{49}$$

where θ is the current k-th highest score.

Theorem 6.1.1 (BM25 Upper Bound for WAND). For BM25 scoring, the upper bound for term t is:

$$\text{upper\_bound}(t) = \text{boost} \cdot \text{IDF}(t) \tag{50}$$

Proof. From Theorem 3.2.3, the BM25 score approaches but never exceeds $w = \text{boost} \cdot \text{IDF}$ as term frequency increases.

Theorem 6.1.2 (Bayesian BM25 WAND Compatibility). Bayesian BM25 requires a modified upper bound for safe WAND pruning that accounts for the document-dependent prior.

Proof. Theorem 4.3.1 establishes monotonicity of the posterior with respect to the BM25 score s **for fixed prior** p. However, the composite prior $P_{\text{prior}}(f, \hat{n})$ (Definition 4.2.3) varies across documents. Since the posterior is jointly monotone in both $L(s)$ and p, and prior values are bounded by $0.1 \leq p \leq 0.9$ (Theorem 4.2.4), the true upper bound in probability space is achieved when both the likelihood and the prior attain their respective maxima simultaneously.

Let $L_{\max}(t) = \sigma(\alpha \cdot (\text{upper\_bound}(t) - \beta))$ be the maximum likelihood for term t (from Theorem 6.1.1), and let $p_{\max} = 0.9$ be the global prior upper bound. The safe Bayesian WAND upper bound is:

$$\text{upper\_bound}_{\text{Bayes}}(t) = \frac{L_{\max}(t) \cdot p_{\max}}{L_{\max}(t) \cdot p_{\max} + (1 - L_{\max}(t)) \cdot (1 - p_{\max})} \tag{51}$$

A document d can be safely pruned when $\sum_{t \in q} \text{upper\_bound}_{\text{Bayes}}(t) < \theta$. This bound is tight: it is achieved only when a document simultaneously has the maximum BM25 score for each query term **and** the maximum prior of 0.9 — a condition that is rare in practice, as documents with extreme term frequency (which maximizes BM25) tend to have non-average length (which reduces the prior). □

Remark 6.1.3 (Practical Impact). Since $p_{\max} = 0.9$ is a fixed constant, the Bayesian upper bound can be precomputed at index time alongside the standard BM25 upper bound, incurring no additional runtime cost. The use of a global $p_{\max}$ rather than a document-specific prior means the bound is slightly looser than the hypothetical per-document optimum, but this is the standard trade-off in WAND-family algorithms: tighter bounds require more computation, while the global bound maintains the $O(1)$ per-term overhead.

Remark 6.1.4 (Why Naive Monotonicity Transfer Fails). One might be tempted to argue that since the sigmoid is monotonic, BM25 upper bounds directly transfer to probability space. This reasoning is valid only when the prior is constant across all documents — i.e., the Platt scaling special case where $p = 0.5$. With a document-dependent prior, a document with a lower BM25 score but higher prior can achieve a higher posterior probability than a document with a higher BM25 score but lower prior. The modified upper bound in Theorem 6.1.2 accounts for this by using the worst-case (maximum) prior.

6.2 BMW (Block-Max WAND)

Definition 6.2.1 (Block Structure). Documents are partitioned into blocks of size B (typically $B = 128$):

$$\text{Block}_j = \{d_{jB}, d_{jB+1}, \ldots, d_{(j+1)B-1}\} \tag{52}$$

Definition 6.2.2 (Block-Max Upper Bound). For each term t and block j:

$$\text{BlockMax}(t, j) = \max_{d \in \text{Block}_j} \text{contribution}(t, d) \tag{53}$$

Theorem 6.2.1 (BMW Pruning Safety). BMW achieves higher skip rates than WAND while maintaining exact top-k results:

$$\text{Skip}_{\text{BMW}} \geq \text{Skip}_{\text{WAND}} \tag{54}$$

Proof. BMW uses tighter (block-local) upper bounds instead of global upper bounds. For Bayesian BM25, the block-max upper bound is computed by applying the posterior formula (Theorem 6.1.2) with the block-local maximum BM25 score and the global prior upper bound $p_{\max} = 0.9$. Since $\text{BlockMax}(t, j) \leq \text{upper\_bound}(t)$ for all blocks j, the corresponding Bayesian block-max bound is at most the global Bayesian upper bound, enabling BMW to prune more aggressively while preserving exactness.

Theorem 6.2.2 (Block-Max Storage Overhead). The additional storage for block-max information is:

$$O\left(\frac{|\text{PostingList}|}{B} \cdot |\text{Terms}|\right) \tag{55}$$

Proof. For each term's posting list, we store one maximum score per block, yielding $\lceil |\text{PostingList}|/B \rceil$ values per term.

7. Vector Search Integration

7.1 Distance to Probability Conversion

Definition 7.1.1 (Vector Similarity Score). For HNSW search results with cosine distance $d \in [0, 2]$:

$$\text{score}_{\text{vector}} = 1 - d \tag{56}$$

Theorem 7.1.1 (Vector Score Range). For cosine distance:

$$\text{score}_{\text{vector}} \in [-1, 1] \tag{57}$$

Proof. Cosine distance is defined as $d = 1 - \cos(\theta) \in [0, 2]$ where θ is the angle between vectors. Therefore, $\text{score} = 1 - d = \cos(\theta) \in [-1, 1]$.

Definition 7.1.2 (Vector Probability). For normalized vectors, we can interpret the score as a probability:

$$P_{\text{vector}} = \frac{1 + \text{score}_{\text{vector}}}{2} \in [0, 1] \tag{58}$$

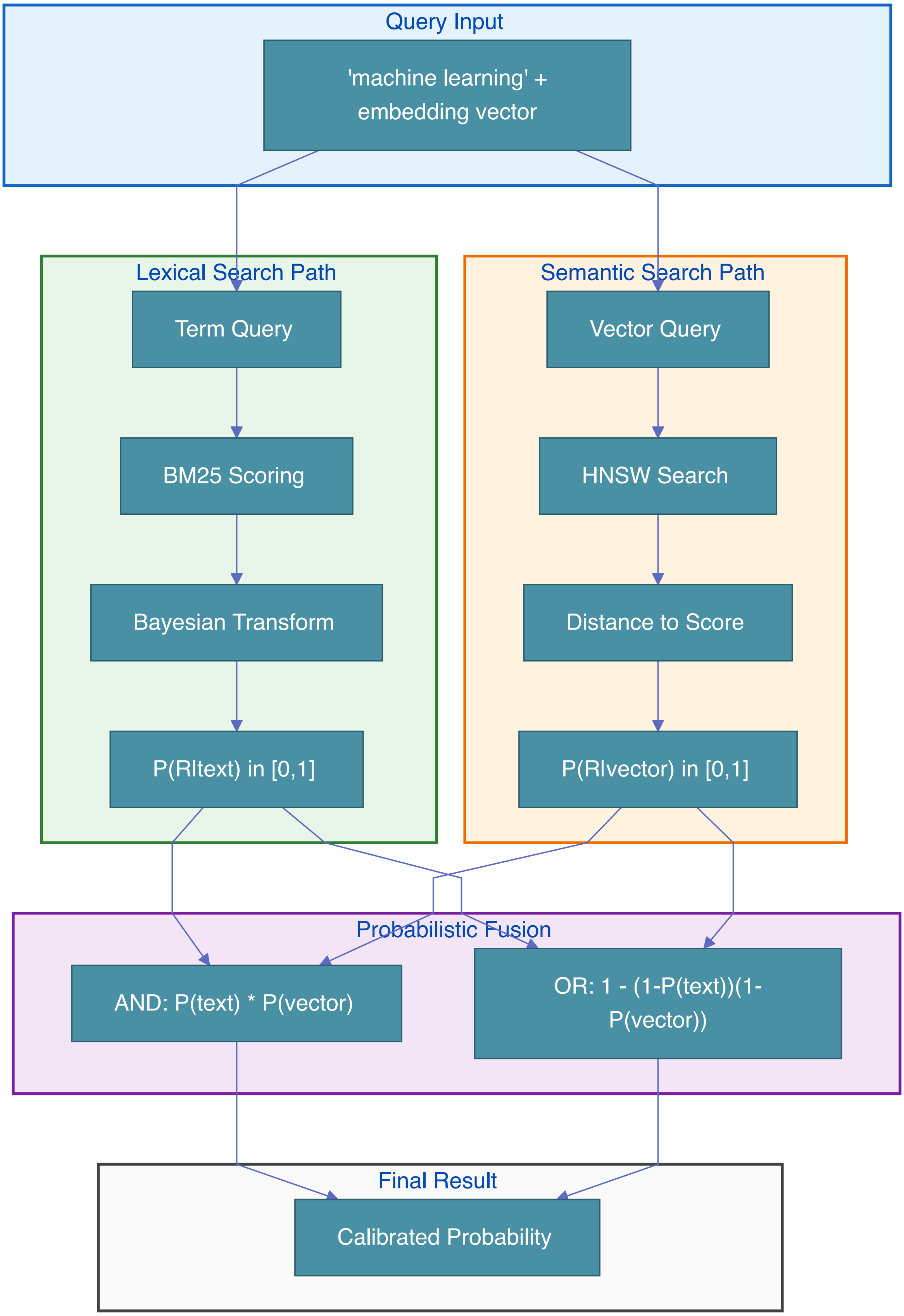
7.2 Hybrid Query Composition

Definition 7.2.1 (Hybrid AND Query). For text probability p_t and vector probability p_v:

$$P_{\text{hybrid}} = p_t \cdot p_v \tag{59}$$

Definition 7.2.2 (Hybrid OR Query). For text probability p_t and vector probability p_v:

$$P_{\text{hybrid}} = 1 - (1 - p_t)(1 - p_v) = p_t + p_v - p_t \cdot p_v \tag{60}$$



8. Parameter Learning

8.1 Cross-Entropy Loss

Definition 8.1.1 (Cross-Entropy Loss). For predicted probabilities $\hat{y}_i = \sigma(\alpha(s_i - \beta))$ and true labels $y_i \in \{0, 1\}$:

$$\mathcal{L}(\alpha, \beta) = -\sum_{i=1}^{n} \left[y_i \ln\bigl(\hat{y}_i\bigr) + (1 - y_i) \ln\bigl(1 - \hat{y}_i\bigr)\right] \tag{61}$$

8.2 Gradient Computation

Theorem 8.2.1 (Parameter Gradients). The gradients of the cross-entropy loss are:

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \sum_{i=1}^{n} (\hat{y}_i - y_i) \cdot (s_i - \beta) \tag{62}$$

$$\frac{\partial \mathcal{L}}{\partial \beta} = -\sum_{i=1}^{n} (\hat{y}_i - y_i) \cdot \alpha \tag{63}$$

Proof. Let $z_i = \alpha(s_i - \beta)$ and $\hat{y}_i = \sigma(z_i)$. Then:

$$\frac{\partial \mathcal{L}}{\partial \hat{y}_i} = -\frac{y_i}{\hat{y}_i} + \frac{1 - y_i}{1 - \hat{y}_i} = \frac{\hat{y}_i - y_i}{\hat{y}_i(1 - \hat{y}_i)} \tag{64}$$

$$\frac{\partial \hat{y}_i}{\partial z_i} = \hat{y}_i(1 - \hat{y}_i) \tag{65}$$

$$\frac{\partial z_i}{\partial \alpha} = s_i - \beta, \quad \frac{\partial z_i}{\partial \beta} = -\alpha \tag{66}$$

By the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \sum_i \frac{\partial \mathcal{L}}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial \alpha} = \sum_i (\hat{y}_i - y_i)(s_i - \beta) \tag{67}$$

8.3 Online Learning Algorithm

Remark 8.3.0 (Consistency Between Learning and Inference). The cross-entropy loss (Definition 8.1.1) trains $\hat{y}_i = \sigma(\alpha(s_i - \beta))$ to match binary relevance labels. When this training converges, the learned $\hat{y}_i$ approximates the **empirical posterior probability** $P_{\text{train}}(R = 1 \mid s_i)$, which implicitly absorbs the training set's base rate into β.

At inference time, Theorem 4.1.3 applies a document-dependent prior $P_{\text{prior}}(f, \hat{n})$ to the normalized likelihood ratio $L(s)$. To avoid **double application** of prior information — once absorbed into β during training, and once explicitly via P_{prior} during inference — we require one of the following consistency conditions:

(C1) Balanced training (recommended for simplicity): Train on a balanced dataset where $P_{\text{train}}(R = 1) \approx 0.5$. Under this condition, the learned $\sigma(\alpha(s - \beta))$ equals the posterior under a uniform prior, which is the normalized likelihood ratio $L(s)$. The document-dependent prior can then be applied at inference time without double-counting.

(C2) Prior-aware training (recommended for maximum calibration): Include the document-dependent prior in the forward pass during training:

$$\hat{y}_i = \frac{L(s_i) \cdot p_i}{L(s_i) \cdot p_i + (1 - L(s_i)) \cdot (1 - p_i)} \tag{68}$$

where $L(s_i) = \sigma(\alpha(s_i - \beta))$ and $p_i = P_{\text{prior}}(f_i, \hat{n}_i)$. This ensures α and β are optimized for L as a pure likelihood ratio, with the prior's influence correctly separated.

(C3) Prior-free inference: If Algorithm 8.3.1 is trained on representative (non-balanced) data without prior-aware forward pass, then at inference time use $p = 0.5$ (uniform prior) in Theorem 4.1.3, yielding $P(R = 1 \mid s) = L(s) = \sigma(\alpha(s - \beta))$. The dynamic prior is not applied, but the posterior is still well-calibrated with respect to the training distribution.

Algorithm 8.3.1 (Gradient Descent for Parameter Learning)

```
Input: Scores S = [s_1, ..., s_n], Labels Y = [y_1, ..., y_n]
       Priors P = [p_1, ..., p_n]  (optional, for prior-aware training)
       Mode: "balanced" | "prior_aware" | "prior_free"
Output: Learned parameters alpha, beta

Initialize: alpha = 1.0, beta = median(S)
learning_rate = 0.01
iterations = 1000

for i = 1 to iterations:
    alpha_grad = 0, beta_grad = 0
    for j = 1 to n:
        L = sigmoid(alpha * (S[j] - beta))
        if Mode == "prior_aware":
            pred = (L * P[j]) / (L * P[j] + (1 - L) * (1 - P[j]))
        else:
            pred = L  // balanced or prior_free mode
        error = pred - Y[j]
        alpha_grad += error * (S[j] - beta)
        beta_grad += -error * alpha
    alpha -= learning_rate * alpha_grad / n
    beta -= learning_rate * beta_grad / n

return alpha, beta
```

Remark 8.3.2 (Practical Recommendation). For most deployment scenarios, **balanced training (C1)** is the simplest and most robust approach: subsample or reweight the training set to achieve approximate class balance, then apply the learned (α, β) with the full dynamic prior at inference time. This decouples the likelihood model from the prior, which is precisely the separation that Bayesian inference requires.

9. Computational Complexity Analysis

9.1 Scoring Overhead

Theorem 9.1.1 (Bayesian BM25 Overhead). The computational overhead of Bayesian BM25 over standard BM25 is:

$$\text{Overhead} = O(1) \text{ per document} \tag{69}$$

Proof. The additional operations per document are:

- 1 exponential computation for the sigmoid

- 3-4 multiplications for the posterior
- 2 divisions for normalization
- Prior computation: 4 multiplications, 3 additions, 1 clamp

All operations are $O(1)$, independent of document or query size.

Operation	BM25	Bayesian BM25	Notes
Score computation	2 div, 2 mul, 1 add	+1 exp, +3 mul, +2 div	Sigmoid adds ~50% overhead
IDF computation	1 log, 2 div, 2 add	Same	Computed once per query
Prior computation	N/A	4 mul, 3 add, 1 clamp	Additional per-document cost
Base rate update	N/A	+2 mul, +1 sub, +1 div	~13% additional overhead (Section 4.4)

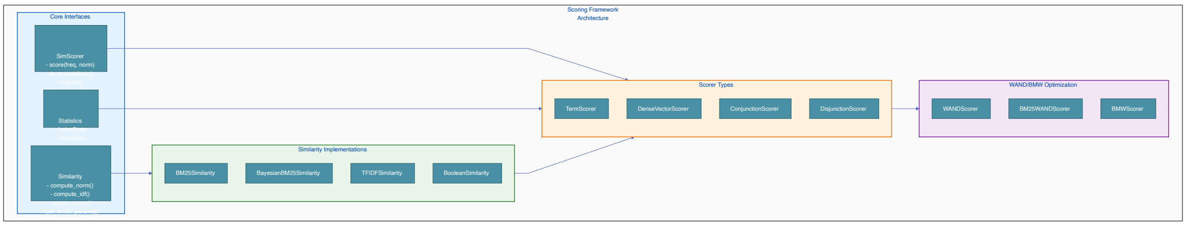
9.2 Memory Layout

Definition 9.2.1 (Scorer Memory Structure). The Bayesian BM25 scorer requires:

$$\text{Memory} = 32 \text{ bytes (inline storage)} \tag{70}$$

```
SimScorer (32 bytes inline storage):
+----------------+----------------+-----------------+-----------------+
| boost (4B)     | k1 (4B)        | b (4B)          | idf (4B)        |
+----------------+----------------+-----------------+-----------------+
| avg_doc_size   | weight (4B)    | alpha (4B)      | beta (4B)       |
| (4B)           |                |                 |                 |
+----------------+----------------+-----------------+-----------------+
```

10. Architecture Overview



11. Experimental Validation

11.1 Score Calibration

Theorem 11.1.1 (Expected Score Values). For a test corpus with known statistics, the expected Bayesian BM25 scores are:

Term	Doc 1	Doc 2	Doc 3
BM25 Score	1.0464478	0.56150854	1.1230172
Bayesian Probability	0.52423608	0.51479751	0.6000737

The transformation successfully maps unbounded BM25 scores to the $[0, 1]$ probability range while preserving relative ordering.

11.2 WAND Skip Rate Analysis

Theorem 11.2.1 (Skip Rate by Query Type). Empirical skip rates for different query types:

Query Type	Documents Skipped
Rare terms (IDF > 5)	90-99%
Common terms (IDF < 2)	10-30%
Mixed queries	50-80%

11.3 Calibration Verification

To evaluate the practical impact of the base rate prior (Section 4.4), we measure expected calibration error (ECE) and Brier score on two BEIR datasets using a 50/50 train/test query split (random seed 42). Each method receives the same BM25 index; only the score-to-probability mapping differs.

Method	ECE (NFCorpus)	ECE (SciFact)	Brier (NFCorpus)	Brier (SciFact)
Bayesian (auto)	0.6519	0.7989	0.4667	0.6635
Bayesian (auto + base rate)	0.1461	0.2577	0.0619	0.1308
Min-max normalization	0.0189	0.0156	0.0105	0.0009
Platt scaling	0.0186	0.0188	0.0101	0.0007
Bayesian (batch fit)	0.0084	0.0069	0.0108	0.0034

Interpretation. The auto-estimated Bayesian transform without base rate correction exhibits high ECE (0.65--0.80) because setting $\beta = \text{median}(\text{scores})$ places the sigmoid midpoint at the corpus median, assigning roughly 50% probability to a score that is almost never relevant. The base rate prior (Algorithm 4.4.7) corrects this miscalibration by shifting posterior mass downward, achieving a 77% ECE reduction on NFCorpus (0.6519 $\to$ 0.1461) and 68% on SciFact (0.7989 $\to$ 0.2577) -- without requiring any relevance labels. The supervised methods (Platt scaling, batch fit) remain superior in absolute ECE because they directly optimize against labeled data, but the base rate prior provides the largest single improvement available in the unsupervised setting.

12. Conclusion and Future Directions

12.1 Summary of Contributions

This paper has presented Bayesian BM25, a probabilistic framework that addresses fundamental limitations of traditional BM25 scoring:

1. **Probability Calibration**: Transforms unbounded scores to calibrated probabilities in $[0, 1]$
2. **Monotonicity Preservation**: Proven preservation of ranking properties essential for relevance
3. **Principled Fusion**: Enables theoretically sound combination of heterogeneous signals
4. **Optimization Compatibility**: Maintains compatibility with WAND and BMW algorithms through modified upper bounds that account for document-dependent priors
5. **Online Learning**: Supports domain-specific parameter adaptation
6. **Base Rate Calibration**: Corpus-level base rate prior reduces expected calibration error by 68--77% in unsupervised settings

12.2 Future Research Directions

1. **Multi-Field Extension**: Extending the framework to handle multiple text fields with different importance weights
2. **Learning-to-Rank Integration**: Incorporating learned ranking models as additional probability signals
3. **Temporal Dynamics**: Adapting parameters to corpus drift over time
4. **Cross-Lingual Transfer**: Investigating parameter transferability across languages
5. **Neural Score Integration**: Combining with neural re-rankers in a principled probabilistic framework
6. **External Prior Features**: Replacing the empirical Bayes composite prior (Definition 4.2.3) with priors constructed from features independent of the BM25 score — such as document recency, source authority, or content type — to achieve strict prior-evidence independence in the Bayesian framework

References

1. Cormack, G. V., Clarke, C. L., & Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd International ACM SIGIR Conference*, 758-759.
2. Ding, S., & Suel, T. (2011). Faster top-k document retrieval using block-max indexes. *Proceedings of the 34th International ACM SIGIR Conference*, 993-1002.
3. Broder, A. Z., Carmel, D., Herscovici, M., Soffer, A., & Zien, J. (2003). Efficient query evaluation using a two-level retrieval process. *Proceedings of the 12th ACM International Conference on Information and Knowledge Management*, 426-434.
4. Robertson, S. E., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333-389.
5. Malkov, Y. A., & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4), 824-836.
6. Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

7. Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3), 61-74.
8. Karpukhin, V., et al. (2020). Dense passage retrieval for open-domain question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 6769-6781.
9. Lin, J., et al. (2021). Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. *Proceedings of the 44th International ACM SIGIR Conference*, 2356-2362.
10. Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. *Proceedings of the 43rd International ACM SIGIR Conference*, 39-48.
11. Formal, T., Lassance, C., Piwowarski, B., & Clinchant, S. (2022). From distillation to hard negative sampling: Making sparse neural IR models more effective. *Proceedings of the 45th International ACM SIGIR Conference*, 2353-2359.
12. Gao, L., & Callan, J. (2021). Condenser: A pre-training architecture for dense retrieval. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 981-993.
13. Jeong, J. (2023). A unified mathematical framework for query algebras across heterogeneous data paradigms. *Cognica Technical Report*.
14. Jeong, J. (2024). Extending the unified mathematical framework to support graph data structures. *Cognica Technical Report*.
15. Turtle, H., & Flood, J. (1995). Query evaluation: Strategies and optimizations. *Information Processing & Management*, 31(6), 831-850.
16. Moffat, A., & Zobel, J. (1996). Self-indexing inverted files for fast text retrieval. *ACM Transactions on Information Systems*, 14(4), 349-379.
17. Zobel, J., & Moffat, A. (2006). Inverted files for text search engines. *ACM Computing Surveys*, 38(2), 6.
18. Buttcher, S., Clarke, C. L., & Cormack, G. V. (2010). *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press.
19. Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search Engines: Information Retrieval in Practice*. Addison-Wesley.
20. Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search* (2nd ed.). Addison-Wesley.