LLM-Assisted Pseudo-Relevance Feedback

David Otero Javier Parapar

CoSCIN'26

IRLab, CITIC, Universidade da Coruña, Spain

UNIVERSIDADE DA CORUÑA

ECIR 2026

Motivation

- IR systems struggle with **vocabulary mismatch**
  - Users' query terms $\neq$ terms in relevant documents

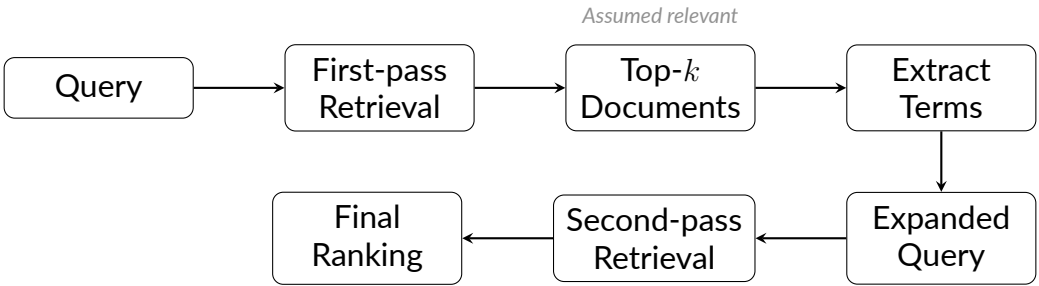
Motivation

- IR systems struggle with **vocabulary mismatch**
  - Users' query terms $\neq$ terms in relevant documents
- **Query Expansion (QE)** addresses this problem
  - Enrich queries with terms that capture user intent

Motivation

- IR systems struggle with **vocabulary mismatch**
  - Users' query terms $\neq$ terms in relevant documents
- **Query Expansion (QE)** addresses this problem
  - Enrich queries with terms that capture user intent
- **Pseudo-Relevance Feedback (PRF)** is a classic QE technique
  - Assumes top-k documents are relevant
  - Extracts expansion terms from them
  - Problem: Some documents may introduce noisy content

What is Pseudo-Relevance Feedback?



- **Key assumption**: Top-k documents from first retrieval are relevant
- Use these documents to find expansion terms

RM3: Relevance-Based Language Model

- A probabilistic approach to PRF
- Estimates a **relevance model** $P(t|R)$ from top-k documents D_k:

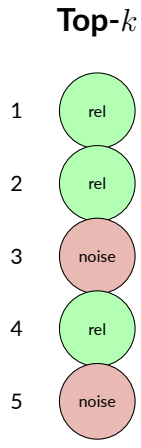
$$P(t|R) \propto \sum_{d \in D_k} P(t|d) \cdot \prod_{w \in q} P(w|d)$$

- **Interpolates** with original query:

$$P_{\text{RM3}}(t) = \lambda \cdot P(t|q) + (1 - \lambda) \cdot P(t|R)$$

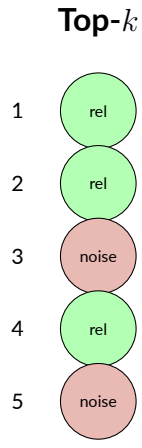
The Problem: Some Documents May Be Noisy

- Top-k may contain **noisy documents**



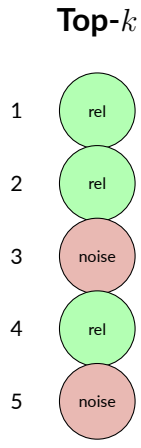
The Problem: Some Documents May Be Noisy

- Top-k may contain **noisy documents**
- Noise $\rightarrow$ irrelevant expansion terms



The Problem: Some Documents May Be Noisy

- Top-k may contain **noisy documents**
- Noise $\rightarrow$ irrelevant expansion terms
- **Topic drift**: expanded query drifts away from user intent



Recent Approaches: LLM-based Query Expansion

- LLMs can generate:
  - Hypothetical documents
  - Query reformulations
  - Query variants

Recent Approaches: LLM-based Query Expansion

- LLMs can generate:
  - Hypothetical documents
  - Query reformulations
  - Query variants
- **Advantages**: Strong semantic understanding

Recent Approaches: LLM-based Query Expansion

- LLMs can generate:
  - Hypothetical documents
  - Query reformulations
  - Query variants
- **Advantages**: Strong semantic understanding
- **Risks**:
  - Hallucinations – generating non-existent terms
  - Misalignment with collection-specific terminology

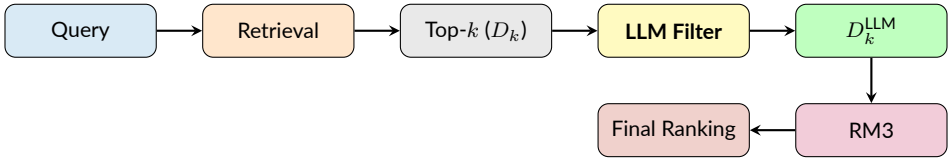
Key Idea

Don’t generate expansion terms with LLMs,

Filter the pseudo-relevant set instead

- Use LLM **judgment capabilities**
- LLM decides which documents in top-k are truly relevant
- RM3 computed only over filtered documents

Method 1: LLM-based Filtering



1. Retrieve top-k documents D_k
2. For each $d \in D_k$: LLM judges relevance $y_d \in \{0, 1\}$
3. Build filtered set: $D_k^{\text{LLM}} = \{d \in D_k : y_d = 1\}$
4. Apply RM3 over D_k^{LLM}

Method 2: Probability-Weighted PRF

- Build on Method 1, but **use LLM confidence**

Method 2: Probability-Weighted PRF

- Build on Method 1, but **use LLM confidence**
- Replace query-likelihood factor with LLM probability:

$$P(t|R) \propto \sum_{d \in D_k^{\text{LLM}}} P(t|d) \cdot P_{\text{LLM}}(\text{'true'}|q, d)$$

Method 2: Probability-Weighted PRF

- Build on Method 1, but **use LLM confidence**
- Replace query-likelihood factor with LLM probability:

$$P(t|R) \propto \sum_{d \in D_k^{\text{LLM}}} P(t|d) \cdot P_{\text{LLM}}(\text{'true'}|q, d)$$

- Documents the LLM is more confident about contribute more

Why This Approach?

LLM-Generation Methods

- Risk of hallucinations
- May introduce terms not in collection
- Less interpretable

Our Approach

- No text generation
- Terms from actual corpus
- Preserves RM3 simplicity

LLM semantic judgment + Corpus-grounded expansion

Experimental Setup

- Collections:
  - AP8889, ROBUST04, MSMARCO (DL-19, DL-20), WT10G
- Compared methods:
  - QLD (baseline), RM3, MonoT5 rerank
  - **MonoT5F**: Filter PRF with MonoT5
  - **LLMF**: Filter PRF with Llama 3.1-8B
  - + variants with probability weighting
- Metrics:
  - AP@1000, NDCG@100

Results

Method	AP88-89	R04	WT10G	DL-20
	AP@1000			
QLD	0.221	0.183	0.152	0.329
+ RM3	0.292	0.200	0.159	0.337
+ MonoT5 rerank	0.247	0.191	0.162	**0.426**
+ MonoT5F + RM3	0.298	**0.232**	0.204	0.394
+ MonoT5F + RM3 w/prob	**0.307**	0.221	**0.207**	0.415
+ LLMF + RM3	0.300	0.214	0.172	0.372
+ LLMF + RM3 w/prob	0.296	0.209	0.175	0.368
+ RM3 oracle	0.384	0.275	0.283	0.435

Ablation: Using TREC Narratives

- TREC topics include **narrative** with relevance guidance
- We augment the LLM prompt with the narrative

	AP@1000			NDCG@100		
Method	AP88-89	R04	WT10G	AP88-89	R04	WT10G
LLMF + RM3	0.334	0.239	0.212	0.495	0.399	0.389
	↑11%	↑12%	↑23%	↑10%	↑10%	↑20%
LLMF + RM3 w/prob	**0.354**	**0.244**	0.206	**0.513**	**0.403**	0.379
	↑20%	↑17%	↑18%	↑14%	↑12%	↑16%
RM3 oracle	0.384	0.275	0.283	0.567	0.454	0.463

Conclusions

- LLM-based **filtering** improves PRF effectiveness
  - Denoises the pseudo-relevant set
  - Avoids hallucination risks
- **Probability weighting** provides additional gains
- Including **narratives** in prompts further improves results
- Simple integration: preserves RM3 well founded simplicity

Future Work

- Detailed analysis of PRF parameter effects
- Exploration of stronger LLM models
- Reasoning-powered alternatives
- Analysis of computational overhead vs. effectiveness trade-offs

Thank you!

We have GitHub repo!

https://github.com/IRLab-UDC/llm-prf