Lost in Decoding? Reproducing and Stress-Testing the Look-Ahead Prior in Generative Retrieval

Kidist Amde Mekonnen **Yongkang Li** **Yubao Tang** **Simon Lupart** **Maarten de Rijke**





IR Lab AMS UNIVERSITY OF AMSTERDAM

Reproduced paper: Zeng, Luo, and Zamani, "**Planning Ahead in Generative Retrieval: Guiding Autoregressive Generation through Simultaneous Decoding**," SIGIR 2024.

SIGIR2026
20-24 July
MELBOURNE | NAARM
AUSTRALIA

In Generative Retrieval, Decoding Can Lose Relevant Documents

Generative Retrieval (GR)

- Each document is represented by a unique docid sequence of length L.
- The model retrieves by generating a docid token by token autoregressively.
- Inference uses **trie-constrained beam search** over valid docid paths.

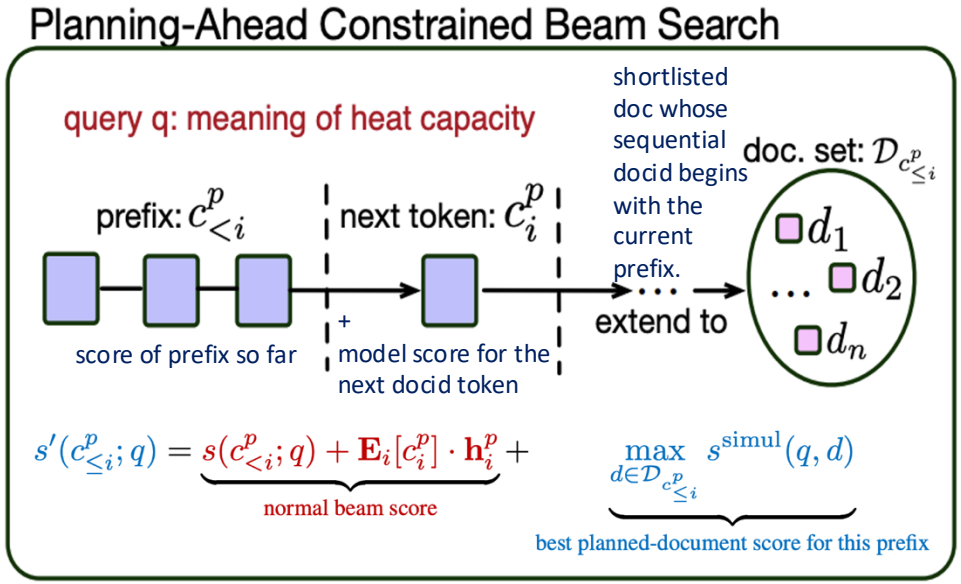
Main challenge: prefix pruning

- Beam search keeps only the top-k prefixes at each step.
- Each document has one valid docid path.
- Once a relevant prefix is pruned, the document is unreachable.

A document is retrieved only if its docid path **survives every decoding step.**

Zeng et al. (2024a): Planning Ahead in Generative Retrieval
Zeng et al. (2024b): Scalable and Effective Generative Information Retrieval
Fig. adapted from: Brenndoerfer, *Beam Search: Decoding and Sequence Generation*, 2025.

PAG Gives Beam Search a Look-Ahead Hint



Adapted from: Zeng, Luo, and Zamani, Planning Ahead in Generative Retrieval, SIGIR 2024.

- PAG runs **simultaneous decoding** alongside standard autoregressive docid generation.
- For each candidate prefix, it estimates the **best document-level relevance score** among the documents reachable from that prefix.
- This look-ahead score is added to the normal beam score, favouring prefixes that can still lead to highly relevant documents.

As a result, prefixes that may lead to relevant documents are less likely to be pruned too early.

PAG uses document-level evidence to guide token-by-token decoding.

PAG: Planning Ahead Through Simultaneous Decoding

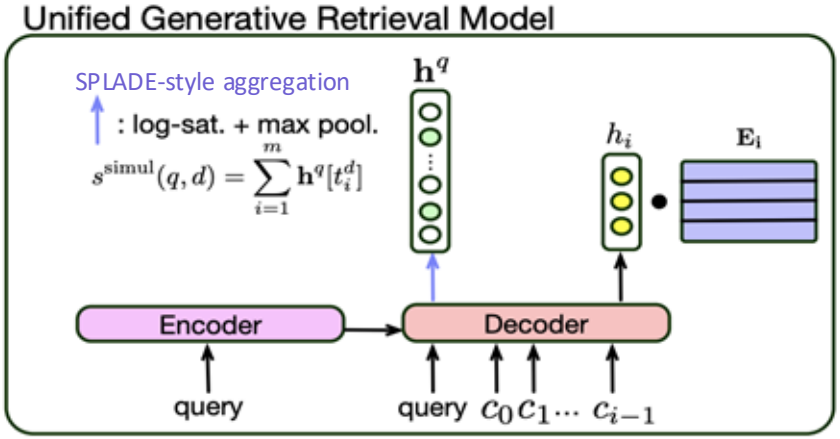


Fig Source: Zeng, Luo, and Zamani, Planning Ahead in Generative Retrieval, SIGIR 2024.

Each document gets two docids:

- Construct new **lexical set-based** docids.
- **Set-based**: Call LM once, then obtain all document-level scores.
- **Lexical:** Provides complementary info. to the original semantic docids.
- Model can be end-to-end trained.

PAG uses the **set-based docid** to score documents and keep the top-n as a candidate pool (paIns). Their scores then guide sequential decoding.

➢ **Planning signal relies on lexical evidence, even small changes in query wording can alter the planning scores and change which documents enter the candidate pool.**

Finding 1: PAG Is Reproducible, and Planning Matters

RQ1–Inference-time validation: To what extent can we reproduce PAG's reported effectiveness and inference-time behavior using the released setup, and what trade-offs emerge across beam sizes?

Main result: At three-decimal precision, reproduced PAG closely matches the reported scores, with a maximum absolute deviation of 0.002.

Effectiveness *Default setting:* $k = 100$, $m = 64$.

Benchmark / metric	Reported	Reproduced
MS MARCO Dev MRR@10	0.385	**0.386**
TREC-DL 2019 NDCG@10	0.705	**0.703**
TREC-DL 2020 NDCG@10	0.700	**0.701**

Reproduced inference-time ablation on MS MARCO Dev

Variant	MRR@10	Recall@10
Full PAG	**0.386**	**0.671**
w/o adding $s_{\text{simul}}(\cdot)$	0.350	0.613
Only $s_{\text{simul}}(\cdot)$ for retrieval	0.303	0.569

Summary: PAG reproduces under the released setup. The planning bonus is effective as **look-ahead guidance** during finite-beam decoding, but planning alone is a weaker retriever.

Finding 2: Effectiveness–Efficiency Trends Mostly Reproduce

tokens for set-based docids

beam size

		MRR@10		Recall@10		Index mem. (GB)		Simul. QL (ms)		Seq. QL (ms)	
m	k	Reported	Reproduced	Reported	Reproduced	Reported	Computed	Reported	Computed	Reported	Computed
16	10	0.342	0.330†	0.577	0.556†	1.30	1.13	20	4	44	42
32	10	0.367	0.360†	0.626	0.608†	1.94	2.26	22	6	44	43
64	10	0.379	0.380	0.641	0.644	3.27	4.53	25	8	44	46
128	10	0.386	—	0.645	—	5.96	—	31	—	44	—
16	100	0.355	0.341†	0.620	0.606†	1.30	1.13	20	4	250	262
32	100	0.372	0.368†	0.652	0.644†	1.94	2.26	22	6	250	263
64	100	0.385	0.386	0.670	0.671	3.27	4.53	25	8	250	266
128	100	0.390	—	0.664	—	5.96	—	31	—	250	—

— denotes settings requiring unreleased artifacts; † marks reproduced effectiveness below the paper.

- **Summary:** At m=64, reproduced scores match the reported results. Larger k improves effectiveness but increases decoding latency; larger m increases planning evidence, memory, and scoring cost.
- **Caveat:** (m=16,32) are truncation-based approximations; (m=128) requires unreleased artifacts.

Finding 3: PAG Is Sensitive to Intent-Preserving Query Variation

RQ2– Robustness & plan drift: How does intent-preserving query variation affect planning stability and downstream ranking?

Same intent, different wording: degradation increases with stronger lexical changes; reordering is comparatively robust.

Variation	MS MARCO Dev MRR@10 (abs. drop)	TREC-DL 2019 NDCG@10 (abs. drop)	TREC-DL 2020 NDCG@10 (abs. drop)
Clean PAG (re-decoded)	**0.386**	**0.700**	**0.702**
Reordering	0.377 (↓0.009)	0.701 (↑0.001)	0.681 (↓0.021)
Naturalizing	0.366 (↓0.020)	0.676 (↓0.025)	0.679 (↓0.023)
Paraphrasing	0.319 (↓0.067)	0.637 (↓0.063)	0.588 (↓0.114)
Synonymizing	0.289 (↓0.097)	0.574 (↓0.126)	0.559 (↓0.143)
Misspelling	**0.246 (↓0.140)**	**0.507 (↓0.194)**	**0.520 (↓0.183)**

Takeaway: Meaning-preserving changes can substantially degrade PAG, with misspellings causing the largest relative drops of 26–36%.

Next: Is this ranking degradation explained by instability in PAG's planning stage?

Finding 4: Performance Drops Coincide with Plan Drift

Lower overlap indicates stronger plan drift: the perturbed query produces a different candidate pool and planner-token set.

	MS MARCO Dev		TREC-DL 2019		TREC-DL 2020	
Variation	**CandOv.**	**TokJac.**	**CandOv.**	**TokJac.**	**CandOv.**	**TokJac.**
Misspelling	**0.312**	**0.336**	**0.374**	**0.352**	**0.348**	**0.363**
Synonym	0.460	0.475	0.439	0.422	0.498	0.485
Paraphrase	0.573	0.536	0.625	0.551	0.569	0.522
Reordering	0.767	0.661	0.801	0.683	0.801	0.684

CandOv. = CandOverlap@100 (top-100 planned documents), TokJac. = TokJaccard@100 (top-100 planner tokens), clean q vs. perturbed $\tilde{q}$. Higher = more stable plan.

Takeaway: Across all three splits, variations with larger ranking drops also show lower plan overlap: misspellings drift most, while reordering drifts least.

Next: RQ3 tests whether this instability becomes more severe under cross-lingual mismatch with a fixed English index.

RQ4 Cross-lingual shift with a fixed English index

Question: How much performance can PAG recover for non-English queries without re-indexing?

Evaluation setup:

- mMARCO queries in Dutch (nl), French (fr), German (de), and Chinese (zh)
- Fixed English MS MARCO collection
- Released set-based docids, sequential docids, and trie
- No document-side changes or re-indexing

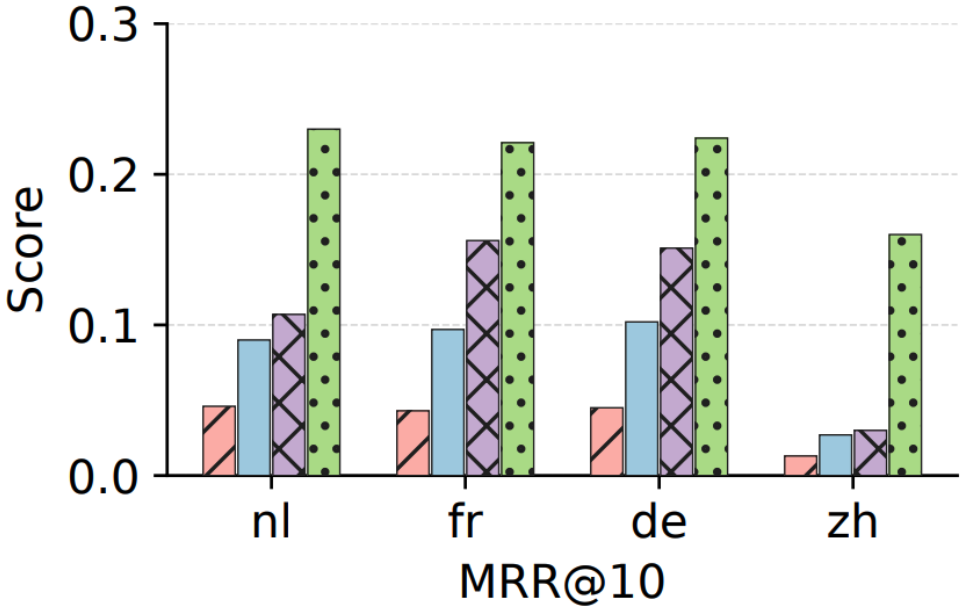
Query-side Settings

- Naive: apply English PAG directly to non-English queries
- Seq-only: disable planning and use sequential decoding only
- Translate: translate queries into English using M2M100
- Aligned: adapt only the query-side planner using aligned English queries

Challenge: The query language differs from PAG's English planning and index space: **Chinese (zh)** introduces script mismatch, **German (de)** and **Dutch (nl)** richer morphology, and **French (fr)** lexical mismatch despite sharing the same script.

Finding 5: Fixed-Index Cross-Lingual Shift Is Harder

- **Naive** PAG remains above Seq-only, suggesting that **weak cross-lingual planning evidence may still preserve useful docid prefixes** through shared entities, numbers, cognates, English terms, or partial lexical overlap.
- **Planner-token distillation** yields only partial gains because improved token alignment does not reliably recover the correct candidate set.
- **Translation** provides the strongest recovery by restoring compatibility with the English planning space.



- **Main message:** PAG shows that planning helps only when the plan stays aligned; more broadly, methods that rely on intermediate guidance should evaluate not only end-to-end performance, but also whether that guidance remains stable and useful under input shift.

Thank You !

Questions?