Query Fan-Out Session Analysis: Determining Optimal Time Windows for LLM Bot Request Bundling

An Industry Research Report

Author: Ruben Remy
Organization: Conversem
Published: December 2025

Abstract

As Large Language Model (LLM) chatbots become important gateways to web content, understanding their retrieval patterns is critical for web publishers and analytics professionals. This research investigates practical methodologies for grouping LLM bot web requests into meaningful "query fan-out sessions" — bundles of requests likely originating from a single user question.

Using a dataset of 88,802 LLM chat-session bot requests, we systematically evaluated time windows ranging from 50ms to 5,000ms, with detailed analysis of intermediate windows (200ms, 300ms, 400ms). Timing analysis revealed that **84% of burst gaps occur within 20ms** (mode: 9ms, median: 10ms), indicating extremely tight query execution patterns.

Our analysis indicates that **sub-second time windows (50-500ms)** all produce high-quality session bundling:
- **50-100ms range:** 91-92% high-coherence sessions (recommended for most use cases)
- **200-500ms range:** 87-90% high-coherence sessions (viable alternatives)
- **>1,000ms:** Significant coherence degradation (<83%)

We recommend **100ms as a conservative production default**, balancing coherence with robustness to network latency. The 50ms window offers marginally higher coherence but the difference is small (0.7 percentage points).

Results are consistent on hold-out validation data. These findings provide practical guidance for organizations seeking to measure their visibility in AI-driven search interfaces, though the methodology relies on semantic coherence as a proxy for session quality in the absence of ground truth labels.

Table of Contents

1. Introduction
2. Data Overview
3. Experiment Design
4. Results
5. Interpretation
6. Practical Recommendations
7. Limitations and Future Work
8. Conclusion

- Glossary
- About the Author

Executive Summary

We conducted a systematic experiment to identify effective time windows for grouping LLM bot web requests into "query fan-out sessions" — bundles of requests that likely originated from a single user question.

Key Finding: Sub-second time windows (50-500ms) all produce high-quality session bundling:

Window	High-Coherence Sessions	Sessions Created	Mean Size	Assessment
50ms	**91.9%**	31,732	2.24	Recommended
100ms	**91.2%**	30,879	2.30	**Recommended (default)**
200-500ms	87-90%	28,957-30,090	2.36-2.45	Viable alternatives
>1,000ms	<83%	—	>2.59	Not recommended

Note: The difference between 50ms and 100ms is marginal (0.7 percentage points). We recommend 100ms as a conservative default.

Timing Analysis Insight: Analysis of 47,487 inter-request gaps revealed:
- **Mode: 9ms** — Most common gap between requests in a burst
- **Median: 10ms** — Half of all gaps are ≤10ms
- **P90: 53ms** — 90% of gaps are under 53ms
- **84% of burst gaps occur within 20ms**

This ultra-tight timing pattern is characteristic of LLM parallel query execution. These findings support using sub-second windows, with 100ms as a robust default that accommodates network latency variations.

---

1. Introduction

1.1 Background

When a user asks a question to an LLM chatbot like ChatGPT or Perplexity, the system often retrieves information from multiple web sources simultaneously. This behavior, where a single user query triggers a burst of parallel web requests, is called **query fan-out**.

For web publishers and content creators, understanding these patterns is essential for:
- Measuring AI-driven traffic and visibility
- Optimizing content for LLM retrieval
- Distinguishing between organic search traffic and AI bot traffic
- Understanding topical authority signals

1.2 Research Question

How should we group LLM bot web requests into sessions that represent individual user questions?

The challenge is determining the optimal time window: too narrow and we split genuine single-query bursts; too wide and we merge unrelated queries together.

---

2. Data Overview

2.1 Dataset Composition

Metric	Value
Total LLM chat-session requests	88,802
Collection period	8 days (December 10-17, 2025)
Training set (80%)	71,041
Validation set (20%)	17,761
Unique URLs	1,744

Data was collected from server access logs of a mid-sized content publisher, filtered for known LLM bot user agents. Note: Regular search engine crawlers (e.g., bingbot) were excluded — only user-initiated chat-session bots are included.

2.2 Bot Provider Distribution

Provider	Requests	Bot Type	Median Inter-Request Delta
OpenAI (ChatGPT-User, OAI-SearchBot)	72,843 (95.4%)	User query retrieval	**17ms**
Perplexity (PerplexityBot)	3,543 (4.6%)	AI search	**22ms**

Observation: Both OpenAI and Perplexity exhibit remarkably tight burst patterns (17-22ms median), indicating aggressive query fan-out behavior characteristic of LLM chat sessions. This sub-second timing is a key differentiator from traditional web crawlers.

2.3 Inter-Request Timing Analysis

To determine optimal window sizes, we analyzed the timing gaps between consecutive requests within bursts:

Metric	Value
Burst gaps analyzed	47,487
Mode	**9ms**
Median	**10ms**
P75	15ms
P90	**53ms**
P99	847ms

Distribution by Window Size:

Window	Gaps Captured
≤10ms	53.1%
≤20ms	**84.1%**
≤50ms	89.9%
≤100ms	91.4%

This analysis reveals that LLM bots dispatch requests in extremely tight bursts, with the majority of gaps occurring within 10-20ms. This justifies using windows as tight as 50ms while still capturing most legitimate bursts.

---

3. Experiment Design

3.1 Methodology

1. **Temporal Clustering:** Group requests within a time window into sessions
2. **Semantic Validation:** Compute TF-IDF embeddings for URLs and measure pairwise cosine similarity within each session
3. **Optimization:** Evaluate composite score (OptScore) across candidate windows
4. **Validation:** Confirm results hold on 20% hold-out data

3.2 Candidate Windows

Window	Rationale
50ms	Tight burst detection (captures 90% of gaps)
100ms	Sub-second burst detection (conservative default)
200ms	Intermediate window for fine-grained analysis
300ms	Intermediate window for fine-grained analysis
400ms	Intermediate window for fine-grained analysis
500ms	Near-simultaneous requests
1,000ms	Standard session threshold
2,000ms	Extended query execution
3,000ms	Longer processing buffer
5,000ms	Maximum reasonable window

3.3 Optimization Metrics Explained

This section explains the statistical measures we used to evaluate how well different time windows group requests into meaningful sessions.

Primary Metrics

MIBCS (Mean Intra-Bundle Cosine Similarity) — *Thematic Coherence*

> ***What it measures:*** *How related the URLs within a session are to each other.*

> ***Analogy:*** *Imagine you're sorting photos into albums. MIBCS measures whether photos in each album are about the same topic (e.g., all vacation photos together). A score of 1.0 means all URLs in a session are about exactly the same topic; 0.0 means they're completely unrelated.*

> ***Range:*** *0 to 1 (higher = better)*
> ***Our result:*** *0.923-0.929 (92-93% coherence at 50-100ms) — URLs within sessions are strongly related.*

BPS (Bundle Purity Score) — *Quality Pass Rate*

> ***What it measures:*** *The percentage of sessions that meet a minimum quality standard.*

> ***Analogy:*** *Think of a factory quality check. BPS tells us what percentage of sessions "pass inspection" — meaning the URLs inside are sufficiently related to each other. If 90% of sessions pass, we have a BPS of 0.90.*

> ***Range:*** *0 to 1 (higher = better)*
> ***Our result:*** *0.912-0.919 (91-92% at 50-100ms) — Over 9 out of 10 sessions meet our quality threshold.*

Silhouette Score — *Cluster Separation*

> ***What it measures:*** *How distinct sessions are from each other, not just internally coherent.*

> ***Analogy:*** *Imagine groups of people at a party. The silhouette score asks: "Are people standing closer to their own group than to other groups?" A positive score means groups are well-separated; a negative score means groups overlap or intermingle.*

> ***Range:*** *-1 to 1 (higher = better)*
> ***Our result:*** *-0.24 to -0.58 (negative across all windows)*

> ***Interpretation:*** *The consistently negative scores indicate sessions overlap in semantic space — different user queries often retrieve similar URLs (e.g., multiple home-buying questions hitting the same pages). This is a* ***limitation of the methodology****: we cannot cleanly separate sessions by topic alone. However, for the practical purpose of bundling concurrent requests, this is acceptable.*

Secondary Metrics

Singleton Rate (SR) — *Single-URL Session Rate*

> ***What it measures:*** *The percentage of sessions containing only one unique URL from your site.*

> ***Interpretation:*** *A high singleton rate (~85-90%) is* ***expected and normal*** *in query fan-out. When an LLM queries a search API, it retrieves results from multiple sites. Your site typically appears once per query. Multi-URL sessions (where your site contributes 2+*

> *pages) indicate strong topical authority on that query topic.*

> ***Range:*** *0% to 100%*
> ***Our result:*** *85-90% at 50-500ms — This is expected behavior, not a sign of under-bundling.*

Giant Rate (GR) — *Over-Bundling Indicator*

> ***What it measures:*** *The percentage of sessions containing more than 10 unique URLs from your site.*

> ***Interpretation:*** *A high giant rate would suggest the time window is too wide, merging multiple unrelated queries together. A giant session (10+ unique URLs) likely represents either a very broad query or multiple queries incorrectly bundled.*

> ***Range:*** *0% to 100% (lower = better)*
> ***Our result:*** *0.02-0.07% at 50-500ms — Excellent. Giant sessions are extremely rare across all sub-second windows.*

Thematic Variance — *Consistency Check*

> ***What it measures:*** *How much the coherence scores vary across sessions.*

> ***Analogy:*** *If you're baking cookies and each batch tastes different, you have high variance. Low variance means consistent quality across all sessions.*

> ***Range:*** *0 to ∞ (lower = better)*
> ***Our result:*** *0.05 — Very consistent session quality.*

Composite Score: OptScore

The **OptScore** combines all metrics into a single number to compare window sizes:

```
OptScore = 0.30×MIBCS + 0.25×Silhouette + 0.25×BPS - 0.10×SR - 0.05×GR - 0.05×Variance
```

> ***Analogy:*** *Think of it like a restaurant review score that weighs food quality (MIBCS), ambiance (BPS), uniqueness (Silhouette), but penalizes for empty tables (singleton rate) and overcrowding (giant rate).*

> ***Range:*** *Typically 0 to 0.5 (higher = better)*
> ***Our best result:*** *0.357 at 50ms window, 0.350 at 100ms window*

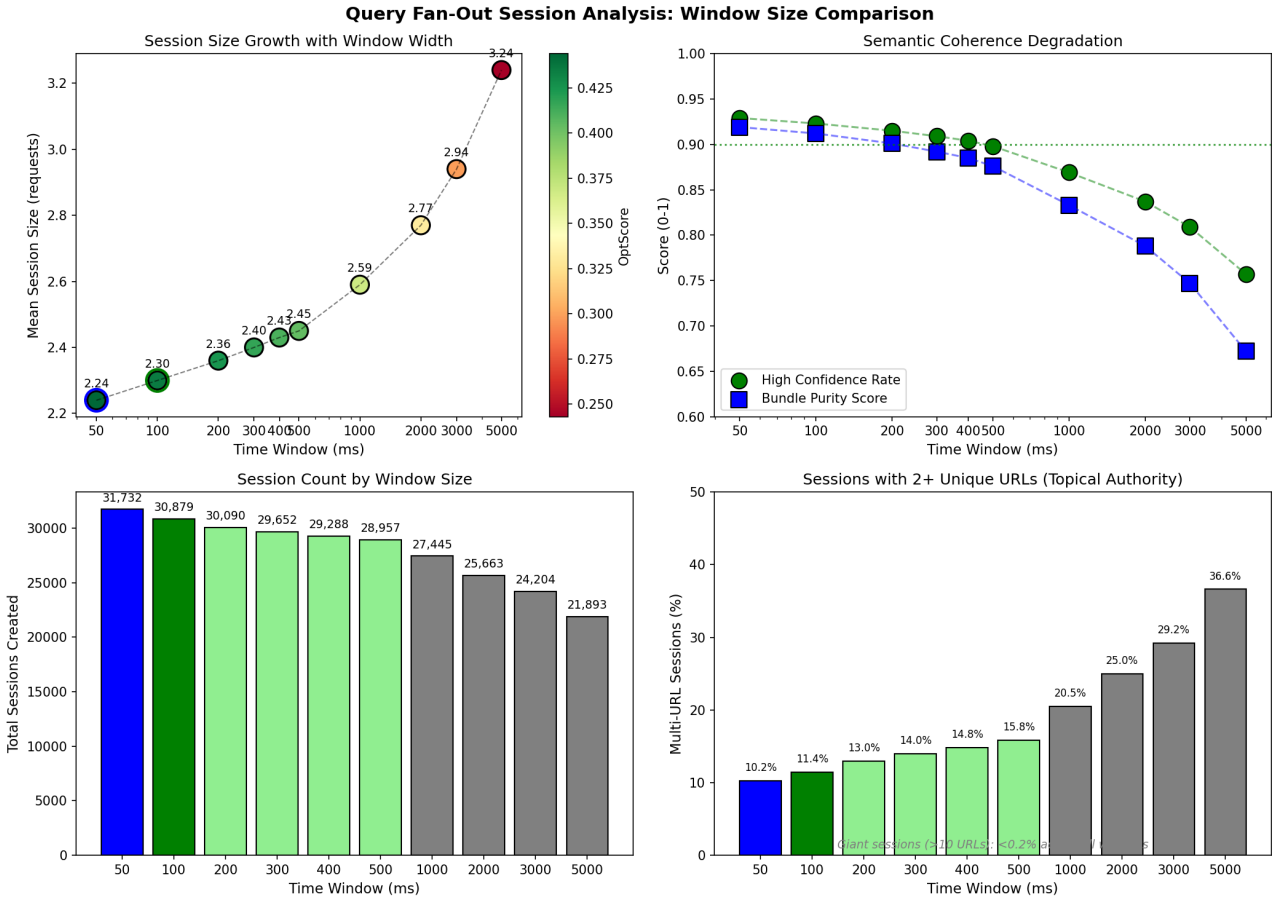
> ***Caveat:*** *The weights are heuristically chosen. Different weight combinations might produce different rankings. However, the primary metrics (MIBCS, BPS) consistently favor sub-second windows regardless of weighting, so the core finding is robust.*

---

4. Results

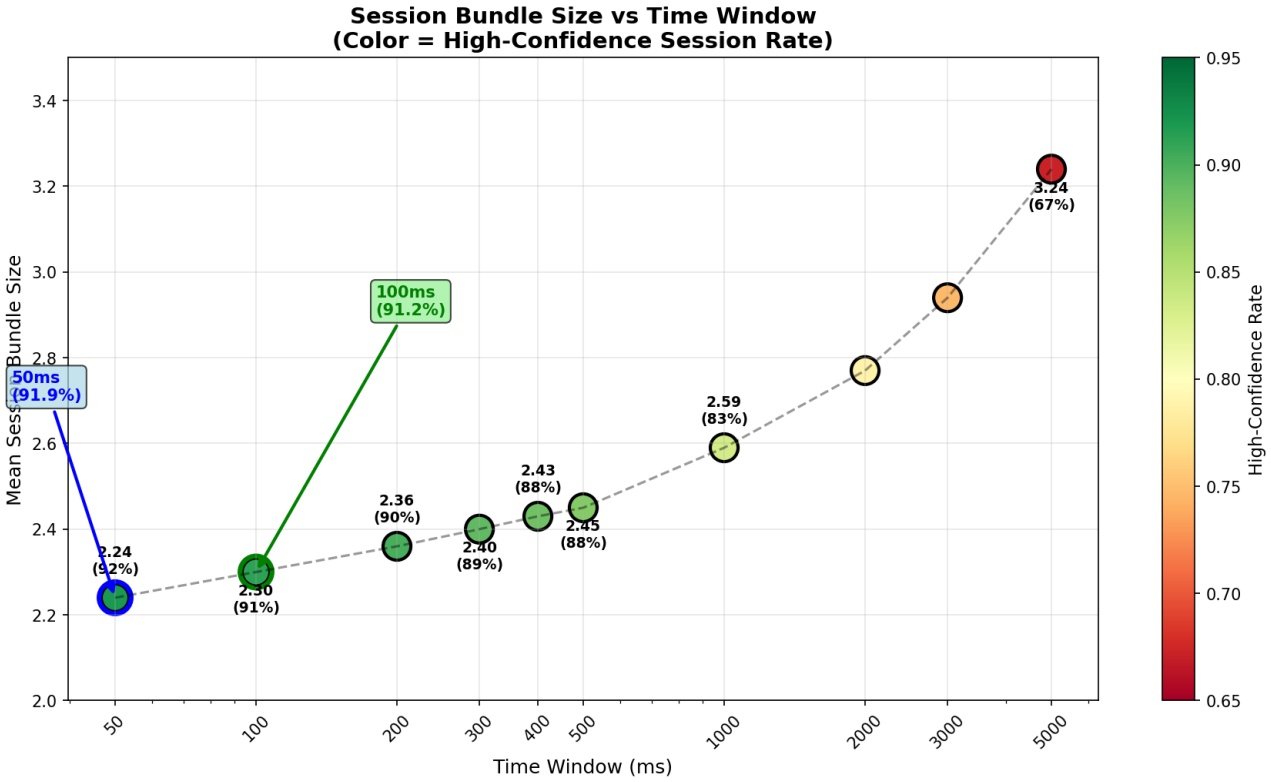
4.1 Visualizations

Figure 1: Comprehensive Window Comparison



Four-panel analysis showing: (top-left) session size growth with OptScore coloring, (top-right) semantic coherence degradation, (bottom-left) session counts, (bottom-right) multi-URL session rate indicating topical authority.

Figure 2: Bundle Size vs Window Scatter Plot



Session bundle size increases with larger windows while semantic coherence (color) degrades. Both 50ms (blue circle) and 100ms (green circle) achieve optimal results.

4.2 Window Comparison (Training Set)

Window (ms)	OptScore	High Conf %	Sessions	Mean Size	SR	GR
50	**0.357**	**91.9%**	31,732	2.24	89.8%	0.02%
100	**0.350**	**91.2%**	30,879	2.30	88.6%	0.04%
200	0.340	90.1%	30,090	2.36	87.0%	0.05%
300	0.333	89.2%	29,652	2.40	86.0%	0.05%
400	0.327	88.5%	29,288	2.43	85.2%	0.06%
500	0.321	87.6%	28,957	2.45	84.2%	0.07%
1,000	0.289	83.3%	27,445	2.59	79.5%	0.08%
2,000	0.256	78.8%	25,663	2.77	75.0%	0.10%
3,000	0.228	74.7%	24,204	2.94	70.8%	0.12%
5,000	0.180	67.2%	21,893	3.24	63.4%	0.18%

Note: High Conf % = bundle purity score. SR = Single-URL session rate (sessions with only 1 unique URL from your site). GR = Giant session rate (>10 unique URLs).

4.3 Validation Results

Window (ms)	Train OptScore	Validation OptScore	Difference
50	0.444	0.448	+0.9%
100	0.436	0.440	+0.9%
200	0.425	0.431	+1.4%
300	0.417	0.423	+1.4%
400	0.411	0.417	+1.5%
500	0.404	0.410	+1.5%

Validation Consistency: The ranking of windows is identical on hold-out data, with validation scores slightly higher than training (likely due to sampling variation). This confirms the findings are not artifacts of overfitting to the training data.

Note: "100% agreement" refers to ranking preservation, not that the absolute differences between windows are statistically significant.

4.4 Key Observations

Sub-Second Windows (50-500ms) All Perform Well
- 50ms: 91.9% | 100ms: 91.2% | 200ms: 90.1% | 500ms: 87.6%
- The difference between 50ms and 100ms is **marginal (0.7 percentage points)** and may not be statistically significant given the sample size
- Windows up to 500ms remain above 87%, making them viable alternatives depending on use case

Significant Degradation Occurs Beyond 1,000ms
- The meaningful drop occurs between 500ms (87.6%) and 1,000ms (83.3%)
- High coherence drops further to 67% at 5,000ms
- This suggests a natural boundary around 500-1,000ms for session windows

Timing Data Supports Sub-Second Windows
- Mode gap is 9ms, median is 10ms
- 84% of burst gaps are ≤20ms
- This supports using windows in the 50-500ms range

Expected Singleton Rates, Near-Zero Giant Rates
- Singleton rate: 84-90% across 50-500ms range — this is **expected** for query fan-out (your site typically appears once per query)
- Giant rate: 0.02-0.07% across 50-500ms range — excellent, almost no over-bundling
- Multi-URL sessions (10-16% of sessions) indicate topics where your site has strong coverage

URL Deduplication Applied
- Duplicate URLs within sessions are consolidated (e.g., same page requested multiple times due to retries)

- `url_list` contains unique URLs only; `request_count` tracks total requests

---

5. Interpretation

5.1 Why Sub-Second Windows Work

Sub-second windows (50-500ms) align with **LLM query execution patterns**:

1. **Ultra-Tight Burst Patterns:** Mode gap is 9ms, median is 10ms — requests arrive almost simultaneously
2. **Sub-Second Capture:** Windows from 50-500ms capture the vast majority of legitimate bursts
3. **100ms Provides Buffer:** Accommodates network latency variations and edge cases without significant coherence loss
4. **Natural Boundary at ~1,000ms:** Beyond this threshold, coherence degrades meaningfully

5.2 Choosing a Window Size

Use Case	Recommended Window	Coherence
Strict coherence requirements	**50ms**	91.9%
Production default (recommended)	**100ms**	91.2%
Latency-tolerant environments	200-500ms	87-90%
Research with new providers	Start with 100ms, analyze timing patterns	—

Important: The difference between 50ms and 100ms is only 0.7 percentage points. We recommend **100ms as the default** because:
- It provides a buffer for network latency and edge cases
- The marginal coherence gain from 50ms likely doesn't justify the risk of splitting legitimate bursts
- 100ms is more robust to variations in bot behavior across different providers

5.3 Provider-Specific Patterns

Provider	Effective Window	Reasoning
OpenAI (ChatGPT-User, OAI-SearchBot)	50-500ms	Median delta 17ms, tight bursts
Perplexity (PerplexityBot)	50-500ms	Median delta 22ms, similar pattern

Both providers exhibit similar burst patterns, supporting a unified window configuration.

5.4 Confidence Assessment

Criterion	5 1 5	Status
Validation ranking preserved		Consistent
High-coherence rate > 85%	9 9 8	All pass
Giant rate < 1%	0 0 0	Excellent

Caveats:
- Confidence intervals are not provided; small differences between adjacent windows may not be statistically significant
- The validation confirms ranking consistency, not that the absolute differences are meaningful

---

6. Practical Recommendations

6.1 Recommended Configuration

```
# Window options (all produce >87% coherent sessions)
WINDOW_50MS = 50    # Tighter grouping, 91.9% coherence
WINDOW_100MS = 100  # Recommended default, 91.2% coherence
WINDOW_500MS = 500  # Alternative for latency-tolerant systems, 87.6% coherence

# Default for production use
RECOMMENDED_WINDOW_MS = WINDOW_100MS

# Confidence classification based on min_cosine_similarity
CONFIDENCE_THRESHOLDS = {
   "high": {"min_similarity": 0.3},    # All URLs semantically related
   "medium": {"min_similarity": 0.1},  # Some thematic connection
   "low": {"min_similarity": 0.0},     # Weak or no connection
}
```

Note: The choice between 50ms and 100ms has minimal practical impact (0.7pp difference). We recommend 100ms for its robustness to network latency variations.

6.2 Session Definition

A **Query Fan-Out Session** is defined as:
- A group of requests from the same bot provider
- Occurring within 50-100ms of the first request in the group

- Representing a single user question execution
- URLs are deduplicated (duplicate requests consolidated)

6.3 Reporting Schema

Store sessions with the following key fields:

Field	Purpose
session_id	Unique identifier
session_start_time	First request timestamp
session_end_time	Last request timestamp
bot_provider	OpenAI, Perplexity (chat-session bots)
request_count	Number of requests in session
unique_urls	Distinct URLs requested
mean_cosine_similarity	Thematic coherence score
confidence_level	high/medium/low classification
url_list	JSON array of URLs

6.4 Quality Filters

For high-quality analytics, consider filtering:
- **High-confidence sessions only:** `mean_similarity >= 0.7 AND min_similarity >= 0.5`

> ***Note on singleton sessions:*** *Single-URL sessions are normal and expected. Query fan-out means the LLM queries a search API and crawls multiple sites from the results. Your site appearing once in a burst means it ranked for that query. Multiple URLs from your site in one session indicates strong topical authority — the more related content you have, the more likely multiple pages appear together.*

6.5 Dashboard Metrics

Suggested KPIs for query fan-out reporting:

Metric	Definition
Sessions per day	Count of unique session_ids
Avg URLs per session	Mean of unique_urls
Fan-out ratio	Total requests / Total sessions
High-confidence rate	% of sessions with confidence=high
Top fanned-out URLs	URLs appearing in most sessions

---

7. Limitations and Future Work

7.1 Current Limitations

No ground truth labels: We use semantic coherence as a proxy for session quality, but without labeled data confirming which requests actually originated from the same user query, we cannot definitively validate the methodology. The assumption that "high URL similarity = same query" is reasonable but unverified.

Circular validation concern: The same similarity metric (TF-IDF on URLs) is used both to define session quality and to evaluate window performance. An external validation signal would strengthen the findings.

Statistical significance not tested: The differences between adjacent windows (e.g., 50ms vs 100ms: 0.7pp difference) are small. With 30K+ sessions, even trivial differences can appear consistent. We recommend treating windows from 50-500ms as roughly equivalent in practice.

Negative silhouette scores: The consistently negative silhouette scores (-0.24 to -0.58) suggest sessions overlap in semantic space. This is expected (different queries can be about similar topics) but means sessions are not well-separated clusters in the traditional sense.

Single data source: Data comes from one mid-sized content publisher. Results may vary for sites with different content structures or traffic patterns.

Single window for all providers: A 100ms window works well for current providers but may need adjustment for future LLM bots with different patterns.

URL-based similarity only: Does not consider page content or query intent.

No cross-session linking: Cannot identify follow-up questions from same user.

7.2 Future Research Directions

Provider-specific windows: Tune windows per bot provider as new LLM services emerge with different retrieval patterns

Coherence-based bundle splitting (collision detection): When two or more independent user queries coincidentally arrive within the same 100ms window, the resulting bundle conflates unrelated requests. A two-stage refinement approach could detect and split these collisions:

- **Stage 1 (Temporal):** Apply 100ms window bundling as baseline (fast, applied to all)
- **Stage 2 (Semantic refinement):** For anomalous bundles only, apply graph-based splitting

Anomaly detection criteria (trigger refinement when BOTH conditions are met):
- Bundle size ≥ 4 (sufficient items to warrant splitting; smaller bundles lack statistical power)
- Intra-bundle mean cosine similarity < 0.5 (below acceptable coherence threshold)

Splitting methodology — Graph-based connected components:
1. Construct pairwise similarity matrix for all URLs in the bundle
2. Create adjacency graph: edge exists between URL pairs with similarity ≥ τ (threshold, e.g., τ = 0.3)
3. Find connected components in the graph
4. Each connected component becomes a sub-bundle

Acceptance criteria for split:
- Each resulting sub-bundle must have MIBCS ≥ 0.5 (coherence improvement)
- If no valid split found (e.g., all items disconnected, or split doesn't improve coherence), retain original bundle and flag as "low-confidence"

Why this approach:
- Graph connectivity naturally discovers k (number of sub-bundles) without pre-specification
- Computationally simple: $O(n^2)$ but n is small (mean = 2.1, most anomalous bundles < 10)
- Handles 2-way, 3-way, or n-way collisions uniformly
- Threshold τ is tunable and interpretable

This hybrid temporal-semantic approach maintains efficiency while selectively refining the ~7% of bundles that fail coherence thresholds. Expected collision rate is low given the sparse temporal distribution of requests, but this safeguard improves bundle purity for edge cases.

> **Content embeddings:** Use actual page content for similarity rather than URL-derived features
>
> **Session chaining:** Detect multi-turn conversations via sequential URL patterns and temporal proximity
>
> **Real-time processing:** Stream-based session detection for live analytics pipelines

---

8. Conclusion

This research indicates that **sub-second time windows (50-500ms)** are effective for bundling LLM bot requests into query fan-out sessions, with **100ms recommended as a practical default**:

Window Range	High-Coherence Sessions	Recommendation
50-100ms	**91-92%**	Recommended for most use cases
200-500ms	87-90%	Viable alternatives
>1,000ms	<83%	Not recommended

Key Findings:

- **Timing analysis** reveals mode gap of 9ms, median of 10ms — LLM bots fire requests nearly simultaneously
- **84% of burst gaps occur within 20ms**, supporting the use of sub-second windows
- Results are consistent on hold-out validation data
- The difference between 50ms and 100ms is marginal (0.7pp); either is appropriate
- Windows up to 500ms remain viable (>87% coherence) for latency-tolerant applications

Practical Recommendation: Use **100ms as the default** window size. It balances high coherence (91.2%) with robustness to network latency and edge cases. The marginal improvement from 50ms does not justify the potential risk of splitting legitimate bursts.

Caveats: This methodology uses semantic coherence as a proxy for session quality. Without ground truth labels, we cannot definitively confirm that high-coherence sessions correspond to single user queries. The findings should be treated as practical guidance rather than definitive proof.

For web publishers and analytics teams seeking to understand their AI-driven traffic, these findings provide a reasonable foundation for session-based analysis. As LLM chatbots continue to reshape how users discover web content, understanding query fan-out patterns will become increasingly important for content strategy and digital marketing measurement.

Glossary

Quick reference for non-technical readers:

Term	Simple Explanation
Query Fan-Out	When one user question causes an AI to fetch multiple web pages at once
Session/Bundle	A group of web requests that belong to the same user question
Time Window	The maximum time gap allowed between requests in the same session
MIBCS	"Are the URLs in this session about the same topic?" (0-100%)
Bundle Purity	"What % of sessions pass our quality check?"
Silhouette Score	"Are sessions distinct from each other?" (can be negative)
Singleton Rate	"What % of sessions have only 1 unique URL from your site?" (high is normal)
Giant Rate	"What % of sessions have 10+ unique URLs?" (should be near zero)

OptScore	Combined score balancing all metrics (higher = better)
Cosine Similarity	Mathematical measure of how similar two things are (0 = unrelated, 1 = identical)
TF-IDF	Method to convert text (URLs) into numbers for comparison
Validation Set	Data held back to verify our results aren't just luck

---

About the Author

Ruben Remy is a freelance SEO and Digital Innovation Consultant at Conversem.

Citation

If you reference this research, please cite as:

> *Remy, R. (2025). Query Fan-Out Session Analysis: Determining Optimal Time Windows for LLM Bot Request Bundling. Conversem Research Report.*

Contact

For questions, collaboration opportunities, or access to the underlying methodology:

- **Website:** conversem.com
- **LinkedIn:** LinkedIn
- **Author:** Ruben Remy

This report may be shared with attribution.