

Lecture 5: Retrieval-Augmented Generation

Lecturer: *Edo Liberty*Scribes: *Hassam Gani*

1 Context, Search, and Retrieval-Augmented Generation

A language model's parameters encode information acquired during training. This information is compressed and does not automatically update when the world changes. Questions about private documents, recent events, or sources that must be cited therefore require access to external information. The lecture asks how to provide this access and how to select useful evidence under a limited context budget.

1.1 Parametric and non-parametric knowledge

Parametric knowledge is stored implicitly in model weights θ . Updating it requires training, for example

$$\theta' \in \arg \min_{\theta} \mathcal{L}(\theta; D_{\text{new}}).$$

Updates can be expensive, interfere with previously learned behavior, and do not guarantee reliable recall or provenance for individual facts.

Non-parametric knowledge is stored explicitly in a corpus $\mathcal{D} = \{d_1, \dots, d_N\}$ of documents, databases, code, or web pages. Relevant information is retrieved and supplied in the context. The corpus can be updated independently of the model's parameters.

1.2 Four ways to access new information

1. **Weight updates.** Continual pre-training, supervised fine-tuning, or adapters such as LoRA modify the model using new data. They avoid repeatedly including that data in the prompt, but require optimization and may leave facts poorly memorized or quickly outdated. In lecture, fine-tuning for this purpose was marked as not recommended, with adapters as its lighter-weight form.
2. **Context stuffing.** Include the full corpus and generate from $p(y \mid x, \mathcal{D})$. This avoids training, but the input may exceed context limits or become expensive and difficult to use.
3. **Static RAG.** Retrieve $S \subseteq \mathcal{D}$ and generate from $p(y \mid x, S)$; the retrieval step is a search problem. Updating an index is typically cheaper than training. However, evidence missed during retrieval may remain unavailable to the generator.
4. **Agentic access.** A policy $\pi(a_t \mid x, h_t)$ selects search, SQL, or API actions based on the query and interaction history. Intermediate observations permit query reformulation and multi-step investigation, with additional latency and token expenditure.

1.3 Why retrieval for an LLM differs from human search

The lecture summarized the difference in two points: the goal is to get *all* the information the answer needs, not the single “right” document; and the consumer is an LLM, not a human. Human search commonly presents an ordered list of individually relevant results. RAG supplies evidence that an LLM uses jointly. Several highly relevant passages may be a poor choice if they all repeat the same information.

For example, five paraphrases of one paragraph may all be relevant, but the last four add little evidence. A complementary passage may be more useful than another high-scoring duplicate. We therefore evaluate a set S through joint utility $f(S)$, rather than only through separate document scores.

Selection and order. A human reader scans results in rank order, so rank largely determines what is seen. An LLM receives the entire selected set, and the lecture treats inclusion as the first-order question: a set objective abstracts away order to study which evidence is included. This does not mean that actual model outputs are invariant to prompt order. Positional effects, including the *lost in the middle* phenomenon (Liu et al., 2024), can make evidence near the beginning or end easier to use. Selection determines what is available; placement can affect how well the model uses it.

Query formulation. A human query may be a short keyword phrase, such as *COMS E6998 Columbia syllabus*. Retrieval can instead use a full question, conversational context, or an embedding. Hypothetical document embeddings (HyDE; Gao et al., 2023) generate a hypothetical answer passage and use its embedding to retrieve actual documents. The generated passage is a search aid, not verified evidence. An agent can also begin broadly with a directory or schema lookup before asking targeted follow-up questions.

Token costs. Each chunk has cost $c(d) = |\text{tokenize}(d)|$. More text increases input-token cost and prefill work. For dense attention, the attention component of prefill scales quadratically with sequence length; KV-cache storage grows linearly for a fixed model. Thus a realistic constraint is

$$\sum_{d \in S} c(d) \leq B.$$

The simpler constraint $|S| \leq k$ models a budget of k chunks (the token budget when all chunks have equal length) and is the setting of the greedy proof below.

1.4 From documents to retrieved context

Documents are split into chunks. An embedding model maps each chunk to a numerical vector, stored in an index with a reference to its original text. The intended geometry places semantically similar chunks near each other. Encoding the query in a compatible space identifies candidates. The retrieved *text*, rather than just its vector, supplies evidence to the generator.

Lexical search matches words or terms; embedding-based search compares vector representations. These approaches can also be combined. In either case, candidates must be selected to fit the context budget. Figure 1 summarizes the pipeline.

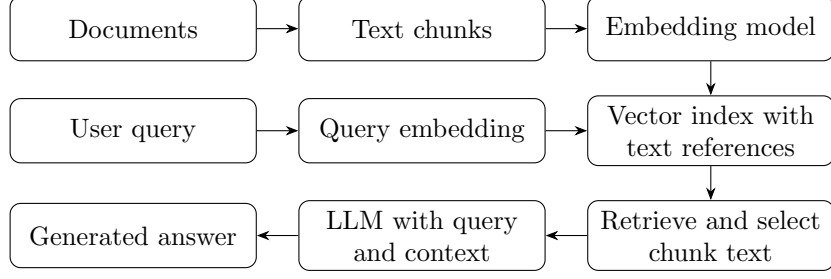


Figure 1: Static RAG. Chunks are embedded offline; at query time the query is embedded into the same vector space, candidates are retrieved from the index, and a selected subset of their text enters the LLM’s context.

2 Maximizing Information as Submodular Optimization

2.1 Subset selection and marginal gain

After retrieval, let $D = \{1, \dots, n\}$ index the candidate chunks and let $f : 2^D \rightarrow \mathbb{R}_{\geq 0}$ measure selected-set utility. The cardinality-constrained problem is

$$\max_{S \subseteq D, |S| \leq k} f(S).$$

For example, retrieval may produce $n = 200$ candidates while only $k = 10$ chunks are allowed. With unequal token costs, the constraint becomes $\sum_{e \in S} c(e) \leq B$.

Definition 1 (Marginal gain). *The marginal gain of adding e to S is*

$$f(S \cup \{e\}) - f(S).$$

This measures additional information after accounting for what is already selected.

Definition 2 (Normalization and monotonicity). *A set function is normalized if $f(\emptyset) = 0$. It is monotone if*

$$S_1 \subseteq S_2 \subseteq D \implies f(S_1) \leq f(S_2).$$

Equivalently, all marginal gains are nonnegative.

These are assumptions on the utility model. They need not hold for every measure of actual answer quality: distracting text can hurt generation. The theorem applies when the chosen objective satisfies the stated assumptions.

2.2 Submodularity and diminishing returns

Definition 3 (Submodularity). *A set function $f : 2^D \rightarrow \mathbb{R}$ is submodular if, for all $S_1 \subseteq S_2 \subseteq D$ and $e \in D \setminus S_2$,*

$$f(S_1 \cup \{e\}) - f(S_1) \geq f(S_2 \cup \{e\}) - f(S_2).$$

Equivalently, for all $A, B \subseteq D$,

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B).$$

The marginal value of a chunk can decrease as more evidence is collected. Information that is new in a small context may already be present in a larger one.

Example 4 (Coverage). *Suppose each chunk contains a set $\text{facts}(e)$ of atomic facts. Then*

$$f_{\text{cov}}(S) = \left| \bigcup_{e \in S} \text{facts}(e) \right|$$

is normalized, monotone, and submodular.

Proof. The empty set covers no facts, and adding chunks cannot remove facts. Moreover,

$$f_{\text{cov}}(S \cup \{e\}) - f_{\text{cov}}(S) = \left| \text{facts}(e) \setminus \bigcup_{a \in S} \text{facts}(a) \right|.$$

For $S \subseteq T$, the set of already-covered facts grows, so the number of new facts supplied by e cannot increase. $\square$

Example 5 (Facility location). *Let V be a finite set of query or topic representations, with $\text{sim}(v, e) \geq 0$ (for instance, cosine similarity clipped at zero). Then*

$$f_{\text{fac}}(S) = \sum_{v \in V} \max_{e \in S} \text{sim}(v, e),$$

where the empty-set maximum is defined as zero, is normalized, monotone, and submodular.

Proof. Write $m_v(S) = \max_{a \in S} \text{sim}(v, a)$. The marginal gain is

$$\sum_{v \in V} \max\{0, \text{sim}(v, e) - m_v(S)\}.$$

As S grows, each $m_v(S)$ increases or stays fixed, so each summand cannot increase. Normalization and monotonicity follow directly. $\square$

These objectives reward complementary evidence while limiting repetition. Submodularity does not model arbitrary synergies where two individually unhelpful chunks become valuable only together.

2.3 The greedy algorithm

Assume $1 \leq k \leq |D|$. Greedy repeatedly chooses the largest current marginal gain:

1. Initialize $S_0 = \emptyset$.
2. For $i = 0, \dots, k-1$, choose

$$e_{i+1} \in \arg \max_{e \in D \setminus S_i} f(S_i \cup \{e\}), \quad S_{i+1} = S_i \cup \{e_{i+1}\}.$$

3. Return S_k .

Maximizing $f(S_i \cup \{e\})$ is equivalent to maximizing $f(S_i \cup \{e\}) - f(S_i)$, since $f(S_i)$ is fixed during the choice. Scores are recomputed relative to the selected set. This avoids repeatedly choosing chunks that were individually useful but have become redundant.

Theorem 6 (Greedy approximation; Nemhauser, Wolsey, and Fisher, 1978). *Let $f : 2^D \rightarrow \mathbb{R}_{\geq 0}$ be normalized, monotone, and submodular. If S^* maximizes $f(S)$ subject to $|S| \leq k$, then*

$$f(S_k) \geq \left[1 - \left(1 - \frac{1}{k}\right)^k\right] f(S^*) \geq \left(1 - \frac{1}{e}\right) f(S^*).$$

Proof. By monotonicity, extend an optimum to size k and write $S^* = \{o_1, \dots, o_k\}$. Fix an iteration $i \in \{0, \dots, k-1\}$.

Step 1: Compare with the union. Monotonicity implies

$$f(S^*) - f(S_i) \leq f(S_i \cup S^*) - f(S_i).$$

Step 2: Add optimal elements one at a time. Define the increments

$$\delta_j = f(S_i \cup \{o_1, \dots, o_j\}) - f(S_i \cup \{o_1, \dots, o_{j-1}\}).$$

Their sum telescopes:

$$f(S_i \cup S^*) - f(S_i) = \sum_{j=1}^k \delta_j.$$

Step 3: Apply diminishing returns. Each set to which o_j is added contains S_i . If $o_j \notin S_i$, submodularity gives

$$\delta_j \leq f(S_i \cup \{o_j\}) - f(S_i);$$

if $o_j \in S_i$, both sides are zero. Therefore

$$f(S_i \cup S^*) - f(S_i) \leq \sum_{j=1}^k [f(S_i \cup \{o_j\}) - f(S_i)].$$

Each term is zero or the gain of an available element. The largest available gain is nonnegative, and greedy obtains it. Consequently,

$$\begin{aligned} f(S^*) - f(S_i) &\leq k \max_{e \in D \setminus S_i} [f(S_i \cup \{e\}) - f(S_i)] \\ &= k(f(S_{i+1}) - f(S_i)). \end{aligned}$$

Step 4: Bound the progress. Rearranging gives the one-step progress inequality

$$f(S_{i+1}) - f(S_i) \geq \frac{1}{k} (f(S^*) - f(S_i)). \quad (1)$$

Each step closes at least a $1/k$ fraction of the remaining gap.

Step 5: Unroll the recurrence. Let $\Delta_i = f(S^*) - f(S_i)$. Then

$$\begin{aligned} \Delta_{i+1} &= \Delta_i - (f(S_{i+1}) - f(S_i)) \\ &\leq \left(1 - \frac{1}{k}\right) \Delta_i. \end{aligned}$$

By normalization, $\Delta_0 = f(S^*) - f(\emptyset) = f(S^*)$. Multiplying the inductive hypothesis by $1 - 1/k \geq 0$ preserves it, so induction gives

$$\Delta_k \leq \left(1 - \frac{1}{k}\right)^k f(S^*).$$

Thus

$$f(S_k) \geq \left[1 - \left(1 - \frac{1}{k}\right)^k\right] f(S^*).$$

Finally, $1 - x \leq e^{-x}$ implies $(1 - 1/k)^k \leq e^{-1}$, proving the bound. $\square$

Why the argument works. The optimal k chunks can close the current gap when added together. Submodularity bounds their combined gain by the sum of individual gains at the current set. One gain must be at least the average, and greedy chooses a gain at least that large. Repetition gives geometric decay of the gap. Figure 2 shows the picture from lecture.

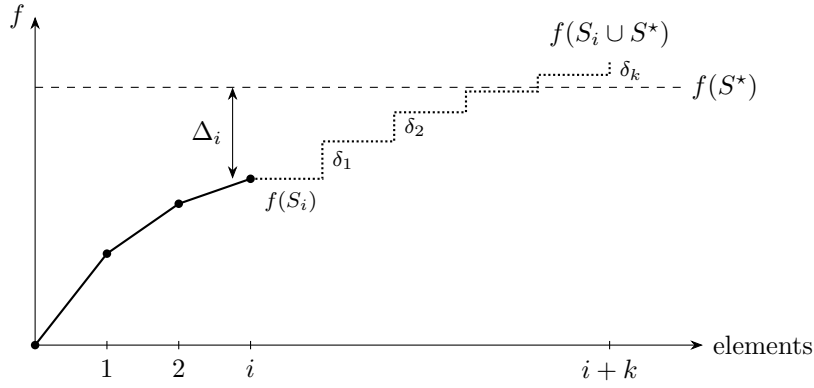


Figure 2: Appending $o_1, \dots, o_k$ to S_i adds the increments δ_j defined in Step 2. Their sum is at least the gap Δ_i . Some δ_j is at least Δ_i/k ; by submodularity so is $f(S_i \cup \{o_j\}) - f(S_i)$, and greedy's next gain is at least as large.

Tightness and token budgets. For general coverage objectives, the $1 - 1/e$ factor cannot be improved by a fixed positive constant in polynomial time unless $P = NP$ (Feige, 1998). Unequal costs require different analysis. Choosing the largest gain per token alone can perform poorly: a cheap chunk can prevent a more valuable chunk from fitting. With budget $B > 2$, disjoint chunks of costs 1, B and values 2, B cause density greedy to obtain only $2/B$ of the optimum.

For monotone submodular maximization under a knapsack constraint, taking the better of density greedy (which adds, at each step, the highest gain-per-token chunk that still fits) and the best feasible singleton guarantees a $\frac{1}{2}(1 - 1/e)$ fraction of the optimum (Khuller, Moss, and Naor, 1999, for coverage; Krause and Guestrin, 2005, in general). Enumerating all seed sets of up to three elements and completing each with density greedy recovers $1 - 1/e$ (Sviridenko, 2004). The cardinality proof does not itself establish these results.

2.4 Running greedy for more than k steps

The lecture closed with a question: what does greedy guarantee if it runs for $k' = k \log(1/\epsilon)$ steps instead of k ? The proof compares each greedy step with an optimum containing at most k elements. This comparator can remain fixed even when greedy selects more elements.

Corollary 7 (Additional greedy steps). *Under the same assumptions, let S^* be optimal under $|S^*| \leq k$. For every $1 \leq t \leq |D|$, after t greedy steps,*

$$f(S_t) \geq \left[1 - \left(1 - \frac{1}{k}\right)^t\right] f(S^*) \geq (1 - e^{-t/k})f(S^*).$$

For $0 < \varepsilon < 1$, choosing

$$t = k' = \left\lceil k \ln \frac{1}{\varepsilon} \right\rceil$$

gives $f(S_{k'}) \geq (1 - \varepsilon)f(S^*)$. If $k' > |D|$, greedy selects all of D , and $f(D) \geq f(S^*)$.

Proof. Steps 1–3 of the proof used only that S^* has at most k elements; they never used $i < k$. Hence (1) holds for every $i < |D|$. Once $i > k$, greedy may already exceed $f(S^*)$, so Δ_i can be negative. Inequality (1) and its rearrangement $\Delta_{i+1} \leq (1 - 1/k)\Delta_i$ remain valid for either sign, and multiplying by $1 - 1/k \geq 0$ preserves them. Induction therefore gives

$$\Delta_t \leq \left(1 - \frac{1}{k}\right)^t f(S^*).$$

Rearrange and apply $(1 - 1/k)^t \leq e^{-t/k}$. For the proposed k' ,

$$k'/k \geq \ln(1/\varepsilon) \implies e^{-k'/k} \leq \varepsilon.$$

If greedy exhausts D , monotonicity gives $f(D) \geq f(S^*)$. □

This is a **bicriteria guarantee**: a larger selection budget lets us approach the best k -element value. The comparator remains the best k -element set, not the best k' -element set; against the latter, Theorem 6 applied with budget k' still gives only $1 - 1/e$. The improvement comes from using more elements, not from a better ratio at the same budget, so it does not conflict with Feige’s bound. The logarithm must be natural (or of any base in $(1, e]$): with $k \log_{10}(1/\varepsilon)$ steps the bound $1 - e^{-t/k}$ gives only $1 - \varepsilon^{1/\ln 10} \approx 1 - \varepsilon^{0.43}$.

For example, $k = 10$ and $\varepsilon = 0.1$ give

$$k' = \lceil 10 \ln 10 \rceil = 24.$$

Selecting 24 chunks guarantees at least 90% of the best value achievable with ten, under the theorem’s assumptions; the sharper factor $(1 - 1/10)^t$ already falls below 0.1 at $t = 22$. These extra chunks also consume context.

3 [Extra/not in class] Beyond a Fixed Budget: Submodular Cover

The extra-steps guarantee suggests a different way to formulate retrieval. Instead of fixing the number of chunks and maximizing their value, fix a desired information level and minimize the resources required to reach it. For example, a query may require evidence about six distinct subtopics, to be covered with as little context as possible.

3.1 From a budget to a target

Let D be finite, let $f : 2^D \rightarrow \mathbb{R}_{\geq 0}$ be normalized, monotone, and submodular, and fix a feasible target $0 < Q \leq f(D)$. The minimum number of chunks needed to attain it is

$$k_Q = \min\{|S| : S \subseteq D, f(S) \geq Q\}.$$

Since $f(\emptyset) = 0 < Q \leq f(D)$, we have $1 \leq k_Q \leq |D|$. Unlike the maximization problem, k_Q is an unknown optimum, not an input budget. We can run an algorithm without knowing it.

Utility above the target is unnecessary for this problem, so we use the *capped objective* $g(S) = \min\{f(S), Q\}$. Then $g(S) = Q$ exactly when S reaches the target.

Lemma 8 (Capping preserves submodularity). *The function g is normalized, monotone, and submodular. Its marginal gain is*

$$g(S \cup \{e\}) - g(S) = \min\{f(S \cup \{e\}) - f(S), Q - g(S)\}.$$

Proof. Normalization and monotonicity follow from those of f and $Q > 0$. If $f(S) \geq Q$, both sides of the identity are zero, since $f(S \cup \{e\}) - f(S) \geq 0$. Otherwise $g(S) = f(S)$, so subtracting $f(S)$ from $\min\{f(S \cup \{e\}), Q\}$ proves the identity. For $S \subseteq T$ and $e \notin T$, submodularity makes the first argument of this minimum no larger at T , and monotonicity makes $Q - g(T) \leq Q - g(S)$. Thus the marginal gain of g cannot increase when S grows, proving submodularity. $\square$

Fix $0 < \varepsilon < 1$. Start with $S_0 = \emptyset$ and repeatedly choose

$$e_{i+1} \in \arg \max_{e \in D \setminus S_i} g(S_i \cup \{e\}), \quad S_{i+1} = S_i \cup \{e_{i+1}\}.$$

Stop as soon as $g(S_i) \geq (1 - \varepsilon)Q$. This rule measures progress against an explicit target; it needs no estimate of k_Q or of the best value achievable with a given number of chunks.

Theorem 9 (Approximate cover with few chunks). *The stopping rule returns S satisfying*

$$f(S) \geq (1 - \varepsilon)Q, \quad |S| \leq \min \left\{ |D|, \left\lceil k_Q \ln \frac{1}{\varepsilon} \right\rceil \right\}.$$

Proof. Until it stops, the rule runs greedy on g , which by Lemma 8 satisfies the assumptions of Corollary 7. Apply the corollary to g with $k = k_Q$. Some set of size k_Q has g -value Q , and $g \leq Q$ everywhere, so the best k_Q -element value of g is exactly Q . After $t = \lceil k_Q \ln(1/\varepsilon) \rceil$ steps, therefore, $g(S_t) \geq (1 - \varepsilon)Q$, and the rule has stopped by then. If $t > |D|$, it stops no later than $S = D$, because $g(D) = Q$. Finally, $f \geq g$. $\square$

With $k = k_Q$, the lecture's extra-steps result becomes a cover guarantee. The benchmark is the smallest *exact* cover; the returned set may miss an ε fraction of the target. Reducing this tolerance tenfold increases the bound by about $k_Q \ln 10$.

Capping is not needed for this count (the corollary applied to f gives the same bound), but it matters once chunks have costs. A capped gain never exceeds the residual $Q - g(S)$, whereas an uncapped gain rewards overshoot: with $Q = 10$, a 50-token chunk covering 1000 facts has uncapped gain per token 20, twice that of a 1-token chunk covering 10 facts, although the short chunk alone meets the target.

3.2 Coverage with unequal token costs

A short chunk and a long chunk should not have identical resource costs. Assign each e a positive cost $c(e)$, write $c(S) = \sum_{e \in S} c(e)$, and define

$$C^* = \min\{c(S) : g(S) = Q\}.$$

Assume now that f is integer-valued and Q is a positive integer, as when f counts covered facts. Costs need not be integers. Starting from $\emptyset$, *density greedy* chooses

$$e_i \in \arg \max_{e \in D \setminus S_{i-1}} \frac{g(S_{i-1} \cup \{e\}) - g(S_{i-1})}{c(e)}$$

until $g(S_i) = Q$. Feasibility and submodularity ensure that some positive marginal exists whenever the target is unmet, so the algorithm terminates.

Theorem 10 (Cost of density greedy). *If S_{gr} is its output, then*

$$c(S_{\text{gr}}) \leq H_Q C^* \leq (1 + \ln Q) C^*, \quad H_Q = \sum_{j=1}^Q \frac{1}{j}.$$

Proof. Let O be an optimal cover of cost C^* . Before step i , let $R_{i-1} = Q - g(S_{i-1}) > 0$, and write $d_i = g(S_{i-1} \cup \{e_i\}) - g(S_{i-1})$. Then $d_i > 0$ as noted above, and $d_i \leq R_{i-1}$ by Lemma 8. Since $g(S_{i-1} \cup O) = Q$, adding the elements of $O \setminus S_{i-1}$ one at a time and applying submodularity, then the greedy density comparison, gives

$$\begin{aligned} R_{i-1} &\leq \sum_{e \in O \setminus S_{i-1}} [g(S_{i-1} \cup \{e\}) - g(S_{i-1})] \\ &\leq \frac{d_i}{c(e_i)} \sum_{e \in O \setminus S_{i-1}} c(e) \leq \frac{d_i}{c(e_i)} C^*. \end{aligned}$$

Hence

$$c(e_i) \leq C^* \frac{d_i}{R_{i-1}}. \tag{2}$$

Integrality has not been used so far. Now $d_i = R_{i-1} - R_i$ with integer residuals, so d_i/R_{i-1} is a sum of d_i copies of $1/R_{i-1}$, and

$$\frac{d_i}{R_{i-1}} \leq \sum_{j=R_i+1}^{R_{i-1}} \frac{1}{j}.$$

The residuals decrease strictly from $R_0 = Q$ to 0, so the blocks $\{R_i+1, \dots, R_{i-1}\}$ are disjoint and together form $\{1, \dots, Q\}$. Summing (2) over all steps yields $c(S_{\text{gr}}) \leq C^* H_Q$. Finally, $\sum_{j=2}^Q 1/j \leq \int_1^Q dx/x$ gives $H_Q \leq 1 + \ln Q$. $\square$

Without integrality a positive residual can be arbitrarily small, and the harmonic charging breaks down for exact cover. Approximate cover still has a guarantee, the weighted counterpart of Theorem 9.

Theorem 11 (Approximate cover with costs). *Drop integrality: let f and Q be as in Section 3.1, and stop density greedy as soon as $g(S) \geq (1 - \varepsilon)Q$, where $0 < \varepsilon < 1$. The output satisfies $c(S) \leq (1 + \ln \frac{1}{\varepsilon})C^*$.*

Proof. Inequality (2) still holds. Let m be the last step. The rule did not stop earlier, so $R_i > \varepsilon Q$ for $i < m$. For these steps, $d_i/R_{i-1} = 1 - R_i/R_{i-1} \leq \ln(R_{i-1}/R_i)$, since $1 - 1/y \leq \ln y$ for $y \geq 1$. The sum over $i < m$ telescopes to $\ln(R_0/R_{m-1}) < \ln(1/\varepsilon)$. The last step has $d_m \leq R_{m-1}$, so it costs at most C^* . $\square$

This covers real-valued objectives such as facility location. For integer utilities run to exact coverage, $R_{m-1} \geq 1$ and the same computation gives $(1 + \ln Q)C^*$: integrality only guarantees that a positive residual is at least one unit.

A worked coverage example. Consider six facts and three chunks $A = \{1, 2, 3, 4\}$, $B = \{1, 2, 5\}$, and $C = \{3, 4, 6\}$, with costs $c(A) = 120$ and $c(B) = c(C) = 100$. Let $f(S)$ count distinct facts and set $Q = 6$. Initial gain-per-cost ratios are $4/120$, $3/100$, and $3/100$, so density greedy first chooses A . Afterwards B adds only fact 5 and C only fact 6. Greedy must take all three chunks, costing 320.

The optimum chooses B, C , covers all six facts, and costs 200. Thus greedy need not be optimal; here its ratio is 1.6, below the bound $H_6 = 49/20 = 2.45$. Under unit costs, the earlier stopping rule with $\varepsilon = 1/6$ would stop after two chunks, covering five facts rather than all six.

The logarithm is necessary. The factor H_Q cannot be improved for density greedy. Take $Q = 6$ facts, a chunk $\{j\}$ for each fact j with cost $60/j$ tokens, and one survey chunk covering all six facts with cost 61. When facts $1, \dots, r$ remain uncovered, the survey has gain per token $r/61$, below the $r/60$ of chunk $\{r\}$. Greedy therefore buys the six short chunks for $60H_6 = 147$ tokens instead of the survey's 61. With $Q \geq 2$ facts and costs $1/j$ and $1 + \delta$, the ratio is $H_Q/(1 + \delta)$ for small $\delta > 0$. No efficient algorithm does much better: set cover is the special case with unit costs, and approximating it within $(1 - \alpha) \ln Q$ is NP-hard for every $\alpha > 0$ (Dinur and Steurer, 2014; earlier, under a stronger assumption, Feige, 1998).

What changes in a RAG system? A target formulation lets the context size adapt to the question: different questions may require different numbers of facts, and repeated evidence contributes no new coverage. With costs equal to token lengths there is no fixed token cap; Theorems 10 and 11 compare the tokens spent with the minimum needed to reach the target, and lowering Q or raising ε relaxes the requirement. Theorem 9 counts chunks and does not bound tokens when lengths differ. All of these guarantees concern the chosen utility model, not the factual correctness of the generated answer.

Relation to the literature. For integer-valued submodular cover with $g(\emptyset) = 0$, Wolsey (1982) proved the sharper bound $H_{\max_{e \in D} g(\{e\})}$ for density greedy ($H_4 \approx 2.08$ in the three-chunk example); the simpler H_Q bound keeps the connection to the lecture's gap argument explicit.

References

- I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proc. 46th ACM Symposium on Theory of Computing (STOC)*, 2014.
- U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

- L. Gao, X. Ma, J. Lin, and J. Callan. Precise zero-shot dense retrieval without relevance labels. In *Proc. ACL*, pages 1762–1777, 2023.
- S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.
- A. Krause and C. Guestrin. A note on the budgeted maximization of submodular functions. Technical Report CMU-CALD-05-103, Carnegie Mellon University, 2005.
- N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *Transactions of the ACL*, 12:157–173, 2024.
- G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.
- M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.
- L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982. <https://doi.org/10.1007/BF02579435>.

AI Use Statement: Used it to verify, correct, refine, and properly cite the presented material, alongside drawing graphs, turning written notes into latex, and cross-checking with Professor Liberty’s official notes for the class