My Proprietary Topic Clustering Engine: A Scoping & Decision Worksheet

JvGLabs |

Helping agencies and their clients become discoverable and recommendable across AI search ecosystems — providing the infrastructure, automation, and expertise to navigate the shift from traditional SEO to AI Visibility.

Introduction

This worksheet is for SEO Directors, Agency Leaders, and Content Strategists who are considering building an in-house, automated topic clustering engine (a system that groups thousands of keywords into meaningful topics using algorithms instead of manual sorting). It solves a specific problem: turning the vague idea of "we should build our own tool" into a concrete, scoped project brief with defined technical requirements, resource needs, and a defensible build-vs-buy decision.

Complete each section in order. Fill in the fields, check the boxes, and use the comparison table to facilitate conversations with your technical team and leadership. By the end, you will have a one-page project scope — not a wish list.

Building this capability is directly tied to AI visibility: as search engines shift from matching keyword strings to understanding topics and entities, owning your own clustering infrastructure means you control how you map, organize, and scale content for discoverability — across both traditional and AI-driven search surfaces.

Quick Win: The 3-Minute Build vs. Buy Litmus Test

Before investing hours in scoping, answer these three questions honestly. They will tell you whether building is even the right conversation to be having right now.

Question 1: Do we have dedicated developer resources (or a budget for them) who understand APIs (application programming interfaces — the connections that let software talk to other software) and data processing?

- ☐ Yes, readily available.
- ☐ Maybe, but they are tied up on other projects.
- ☐ No, we would need to hire or contract this out.

Question 2: Is our primary goal to create a unique, competitive advantage, even if it requires an upfront investment of time and capital?

- ☐ Yes, we want to own a proprietary asset.
- ☐ We are more focused on immediate, low-cost implementation.

Question 3: Will we be processing a high volume of keywords (approximately 100,000 or more) across multiple clients or business units on a regular basis?

- ☐ Yes, our scale justifies a custom solution.
- ☐ No, our needs are project-based and smaller in scale.

How to Read Your Result

If you answered **"Yes" to at least two** of these questions, building a proprietary engine is a strong strategic option. Proceed with the rest of this worksheet.

If you answered "Yes" to only one or none, subscribing to a high-quality third-party clustering tool is likely a more efficient starting point. You can revisit this worksheet when your scale or strategic goals change. There is no shame in buying — the value of this exercise is in making that decision deliberately, not by default.

Section 1: Defining Your Data Inputs

The quality of your clusters depends entirely on the quality of your input data. Garbage in, garbage out. This section forces you to define exactly what raw materials your engine will consume.

1.1 Primary Data Source

Where will your raw keyword data come from? Check all that apply.

- ☐ SEMrush (via API)
- ☐ Ahrefs (via API)
- ☐ Google Keyword Planner (manual export or API)
- ☐ Google Search Console (first-party query data)
- ☐ Other: _____

Important consideration: API access typically requires a paid subscription tier. SEMrush and Ahrefs API pricing varies by plan level and call volume. Confirm your current plan includes API access before committing to a data source. Google Keyword Planner data is free but requires manual export unless you build a connector, which adds development time.

1.2 Data Volume (per project or client)

Be specific. Your developer needs these numbers to estimate processing time and API costs.

- Estimated number of seed keywords (starting terms you will expand from): _____
- Estimated total keywords to cluster after expansion: _____
- Frequency of processing (e.g., monthly, quarterly, per project): _____

1.3 Required Data Points Per Keyword

Check every data point your engine must pull or process. Each additional data point increases API call volume and cost.

- ☐ Keyword text
- ☐ Monthly search volume (approximate; these figures vary by source and are estimates, not exact counts)
- ☐ Keyword difficulty score (a proprietary metric that differs between SEMrush and Ahrefs — they are not directly comparable)
- ☐ SERP features (featured snippets, People Also Ask, etc.)
- ☐ Top 10 ranking URLs for the keyword
- ☐ CPC / commercial value data
- ☐ Search trend data (seasonality)

Section Outcome

You now have a clear definition of the raw materials your engine needs. Write a one-sentence summary of your data setup below.

My engine will pull _____ keywords from _____ on a _____ basis, enriched with the following data points: _____.

Download the Worksheet

Section 2: Choosing Your NLP Model and Clustering Algorithms

This section defines the "brain" of your engine — the models that convert keywords into mathematical representations and the algorithms that group them. Your choice here impacts cost, processing speed, control, and cluster quality.

2.1 NLP / Embedding Model

An embedding model (a system that converts text into numerical vectors so a computer can measure how similar two phrases are) is the foundation. Choose one.

- ☐ **OpenAI Embeddings API** (e.g., text-embedding-ada-002 or newer models): High performance, pay-per-use pricing. Typical cost range is fractions of a cent per thousand tokens (as of the research period — check current OpenAI pricing, as it changes frequently). You send data to an external server.
- ☐ **Google Cloud Natural Language API**: Enterprise-grade, integrates well if you are already on Google Cloud Platform. Pricing is per-request and varies by feature used.
- ☐ **Open-source model** (e.g., Sentence-Transformers library): More complex initial setup. Free to run on your own hardware once configured. Offers maximum control and data privacy — nothing leaves your servers. Requires GPU resources for reasonable processing speed at scale.

Key trade-off: Commercial APIs are faster to implement but create an ongoing dependency. Open-source models require more upfront engineering but eliminate per-call costs and keep data in-house. For 100,000+ keywords, the difference in API costs over a year can be substantial — estimate this before deciding.

2.2 Clustering Algorithm Stack

Most effective pipelines use two steps: first reduce the complexity of the data, then cluster it.

Dimensionality reduction (a technique that simplifies high-dimensional data so clustering algorithms can work more efficiently and accurately):

- ☐ **UMAP** (Uniform Manifold Approximation and Projection — generally recommended for preserving the meaningful structure of the data)
- ☐ **PCA** (Principal Component Analysis — faster but may lose subtle relationships between topics)

Clustering algorithm (the method that actually groups similar keywords together):

- ☐ **HDBSCAN** (Hierarchical Density-Based Spatial Clustering — generally recommended because it does not force every keyword into a cluster, allowing it to identify outliers and produce cleaner topic groups)
- ☐ **K-Means** (simpler, faster, but requires you to specify the number of clusters in advance, which is rarely known beforehand)

A note on "recommended": UMAP + HDBSCAN is the combination most commonly cited in NLP clustering literature for this type of task. However, the best choice depends on your data characteristics and scale. Plan to test both options on a sample dataset before committing. Your developer will need time for this evaluation — typically a few days to a week.

Section Outcome

You now have a technical specification for the core logic of your engine. Summarize it below.

My engine will use _____ for embeddings, _____ for dimensionality reduction, and _____ for clustering.

Section 3: Scoping the Actionable Output

A clustering engine that produces output nobody uses is a waste. This section defines what the finished product looks like and how your team will interact with it daily.

3.1 Final Output Format

How will your team access and use the clustered data? Check your target for Version 1.0.

- ☐ **CSV / Google Sheet export** (simplest to build, lowest development cost, but requires manual handling)
- ☐ **Internal web interface** (e.g., built with Streamlit or Retool — adds a visual layer and basic interactivity, moderate development cost)
- ☐ **Data visualization dashboard** (e.g., Tableau, Looker Studio — best for reporting and stakeholder buy-in, but adds a separate tool dependency)

3.2 Desired Level of Automation

Each level adds development time. Be honest about what you need for Version 1.0 versus what you want eventually.

- ☐ **Level 1 — Clusters only:** The engine groups keywords into topic clusters and outputs them. You do the strategic interpretation manually.
- ☐ **Level 2 — Clusters + pillar identification:** The engine also identifies the primary "pillar" keyword (the broadest, highest-volume term representing each cluster) for each group.
- ☐ **Level 3 — Clusters + pillar identification + automated content briefs:** The engine also generates a structured content brief for each cluster, including pillar/spoke designation (which page is the main hub, which pages support it), suggested H2 headings, and internal linking recommendations. This level typically involves an additional LLM (large language model) call per cluster, which adds both API cost and development complexity.

A realistic note on Level 3: Auto-generated briefs are a starting point, not a finished product. Expect a human editor to review and refine every brief. The time savings come from having a structured draft in minutes instead of hours — not from eliminating human judgment. Factor in the cost of the additional LLM API calls: for large-scale

processing, this can range from modest to significant depending on the model and prompt length.

Section Outcome

Define your Version 1.0 deliverable clearly.

Version 1.0 will deliver Level _____ output in _____ format. Future versions may include: _____.

Section 4: Resource and Cost Estimation

This section helps you prepare for the conversation with your technical team or a contractor. Fill in what you know. Leave blanks for what you need to research — that itself is a useful action item.

4.1 Development Resources

Role	Available In-House?	Estimated Hours for V1.0	Notes
Backend Developer (Python, APIs)	☐ Yes ☐ No	_____	_____
Data Scientist / ML Engineer	☐ Yes ☐ No	_____	_____
Frontend Developer (if building a UI)	☐ Yes ☐ No	_____	_____
Project Manager / SEO Lead (you)	☐ Yes	_____	_____

Typical range for a V1.0 build (Level 1 or 2, CSV output): development time varies widely based on team experience, but plan for approximately 4 to 12 weeks of part-time developer effort as a rough starting range. A Level 3 build with a web interface could take considerably longer. These are approximate figures — get a specific estimate from your developer after sharing this completed worksheet.

4.2 Ongoing Costs

Cost Category	Estimated Monthly Cost	Notes
Keyword data API (SEMrush/Ahrefs)	_____	Check your current plan's API limits

Cost Category	Estimated Monthly Cost	Notes
Embedding API (OpenAI/Google)	_____	Depends on volume; estimate per-run cost
Hosting / Server (if self-hosted model)	_____	GPU instances are more expensive than standard servers
Maintenance / Bug Fixes	_____	Plan for approximately 10–20% of initial build cost per year as a rough benchmark

What most people underestimate: Maintenance is not optional. APIs change their endpoints and pricing. Models get updated or deprecated. Data sources alter their output formats. Budget ongoing developer time for maintenance, or your engine will break within months.

Decision Support: Build vs. Buy Comparison

Use this table to make your final, documented decision. Fill in the "My Situation" column with your honest assessment.

Factor	Building a Proprietary Engine	Buying a SaaS Subscription	My Situation and Priorities
Upfront Cost	High (developer time, typically weeks to months of effort)	Low (monthly subscription, typically in the range of $50–$500/month depending on the tool and tier)	_____
Long-Term Cost	Lower (API usage fees, maintenance)	Higher (subscription costs scale with usage, users, and price increases)	_____
Customization	Full control over algorithms, data sources, output format	Limited to the features the vendor chooses to build	_____
Competitive Edge	High — a proprietary asset your competitors cannot access	None — any competitor can subscribe to the same tool	_____
Speed to Implement	Slow (weeks to months for V1.0)	Fast (same day or same week)	_____

Factor	Building a Proprietary Engine	Buying a SaaS Subscription	My Situation and Priorities
Maintenance Burden	Requires ongoing developer support	Handled by the vendor	_____
Data Privacy	Full control (especially with open-source models)	Data processed on vendor servers	_____

My Decision

Based on this analysis, our path forward is:

- ☐ **Build** — Proceed to share this worksheet with the technical team for a detailed estimate.
- ☐ **Buy** — Select a third-party tool that meets our current needs, and revisit this worksheet in _____ months.
- ☐ **Hybrid** — Start with a third-party tool for immediate needs, and begin scoping a proprietary build in parallel for long-term advantage.

Decision rationale (write 1–2 sentences):

_____

Real-Life Scenario: The "Scope Creep" Trap

This is the single most common way proprietary tool projects fail. Learn from it before you start.

The setup: An SEO Director requests a tool to group keywords into clusters. The development team builds Version 1.0 in six weeks. It works. Then someone asks, "Can it also pull search volume automatically?" Then: "Can we add pillar topic identification?" Then: "Can we build a nice dashboard?" Then: "Can it generate content briefs?" Then: "Can it connect to our CMS?"

What went wrong: Each request is reasonable on its own. But taken together, they transformed a focused clustering tool into an enterprise-grade content platform — with no corresponding increase in budget, timeline, or team size.

The fix: Use this worksheet to define a hard boundary for Version 1.0. Write it down. Get sign-off. Any feature request that falls outside V1.0 scope gets added to a "Version 2.0 backlog" — not squeezed into the current build. Treat each version as a separate project with its own timeline and approval.

Your V1.0 boundary statement (fill in): Version 1.0 will do _____ and nothing else. It will be complete when _____.

Why This Matters for AI Visibility

Search engines and AI systems are moving from matching keyword strings to understanding topics, entities, and semantic relationships. The industry is calling this shift "from strings to things." According to Perplexity Research, 84% of SEO professionals are already automating keyword research, and the AI SEO tools market is projected to grow from approximately $1.8 billion in 2023 to approximately $15.7 billion by 2030 (a CAGR of roughly 28.5%). Building a proprietary topic clustering engine is not a technical exercise — it is infrastructure for staying discoverable and recommendable in AI-driven search ecosystems. Owning this system means you control how your content is mapped, organized, and scaled for the way search actually works now, not how it worked five years ago.

Closing Summary

You now have a documented project scope covering data inputs, technical architecture, output specifications, resource needs, cost estimates, and a clear build-vs-buy decision. This is no longer an idea — it is a brief.

Your completed worksheet is the blueprint. Save it. Print it. Share it with your technical lead.

Your Next Steps

Step 1 — Explore the full strategic guide: This worksheet is one component of a larger framework. Read the complete guide — *Beyond Keywords: The SEO Director's Guide to Building an Automated Topic Clustering Engine* — for the strategic context, technical deep dives, and case study that inform every decision in this document.
→ [Read the full guide on jvglabs.com/topic-clustering-engine-guide]

Step 2 — Brief your technical team: Share this completed worksheet with your lead developer or a qualified contractor. Ask for a time-and-cost estimate based on the specifications you defined in Sections 1 through 4. The clearer your brief, the more accurate their estimate will be.

JvGLabs |

Helping agencies and their clients become discoverable and recommendable across AI search ecosystems — providing the infrastructure, automation, and expertise to navigate the shift from traditional SEO to AI Visibility.

JvGLabs — AI visibility architecture

Download the Worksheet