Working Paper: Automating the Mapping of Value Chains from Open Information Sources

A Scalable Framework for Industrial Capacity Mapping

Justus Schollmeyer*[1], Isabel Estevez[2], Can Kahraman[1], Bruno Danelon[1], Richard Kilea[1], Tobiloba Odeyemi[1], Tom Odhiambo[1], George Boniface[1], Thirunilavan Thavapalan[1], Michael Ndon[1], and Caleb Kalu[1]

[1]RubidiumData
[2]i3T

2025-10-11

Abstract

This working paper presents a method for decomposing products into their underlying value chains using artificial intelligence, specifically large language models (LLMs). Rather than generating data from model memory, we employ LLMs as analytical tools that read, reason, and structure information obtained through automated web search, web scraping, and literature review. Guided by a traceability principle, this approach mimics the workflow of a human researcher: identifying, extracting, summarizing, and cross-referencing data to construct transparent and verifiable value-chain representations. The method proceeds in iterative steps: (0) selecting a value chain anchor product; (1) distinguishing technology types within each system or material; (2) mapping these types to 6-digit HS codes; (3) estimating their prevalence and assigning a prevalence weight; (4) decomposing each type into its constituent parts; and (5) assigning cost-share weights relative to other parts, labor, inputs, and machinery. This recursive decomposition yields a hierarchical technology graph linking technical systems and materials from end products back to raw inputs. The resulting graph supports multiple aggregation levels – from individual technology types and categories to trade-classified HS codes – enabling value share estimation through path-weight calculations at varying levels of granularity. These aggregation layers make it possible to transition seamlessly from engineering-level product structures to trade-level representations of global production systems. We complement this with firm, talent, and intellectual property mappings based on registry, web, and patent data to locate actors within the value chain. Altogether, the framework provides a scalable, transparent, and data-driven approach for reconstructing global value chains from open information sources.

Acknowledgments

This RubidiumData Working Paper builds on an internal discussion paper prepared for the World Bank's FCI Europe and Central Asia division. It benefits from insightful discussions with Jade Sahab (WB), Diana Hristova (WB), Gregor Semieniuk (WB), Emilija Timmis (WB), Lukasz Marek Marc (WB), Andrés Arauz (i3T), Santiago Vásquez (i3T), and Mónica Bruckmann (i3T). The authors alone are responsible for any remaining errors or omissions. The views expressed in this paper do not necessarily reflect those of the individuals acknowledged or their affiliated institutions.

*Corresponding: justus@rubidiumdata.com

Contents

1 Introduction

This working paper presents a method for automating the reconstruction of value chains using large language models (LLMs) in combination with automated web search, and web scraping. The method iteratively decomposes products into types of varying prevalence and, in turn, decomposes each type into parts with varying cost shares, repeating this process until the upstream raw materials are reached. Each product type is mapped to trade codes and other relevant industrial taxonomies.

Because the resulting structure is represented as a graph, it can be analyzed at different levels of aggregation. For example, the 6-digit HS code level provides a more aggregated view than the product level. The high level of product detail also allows firms and intellectual property to be linked directly to their positions within the value chain, without relying solely on abstract classifications such as NACE, ISIC, or HS codes.

The paper is structured as follows: we begin with a general note on how LLMs are used in combination with web search, followed by a description of the algorithmic steps for constructing the technology graph, the mapping of additional information such as firms and intellectual property, the aggregation methods across levels, and finally a discussion of limitations and future work.

2 Value Chain Decomposition Approach

2.1 General note on how we use large language models

We do not generate value chains from a language model's "memory." Instead, we use the model's ability to read, write, summarize, reason, and transform unstructured inputs into structured outputs to mimic the process a human researcher would follow. This process involves conducting literature reviews through automated web search, retrieving relevant sources via lawful and gentle web scraping, excerpting and storing useful information in vectorized form with references to the originals, querying this condensed knowledge as needed, and maintaining full traceability of sources through citations.

We refer to the entire procedure simply as "web search" and to the approach of using language models to transform targeted information that we provide to the model as "traceability principle." This web search draws on publicly available reports, websites, articles, patents, product brochures, and marketplace information (currently exclusively from Alibaba). None of this data is used for training purposes; neither by us nor by third parties. All models we employ are configured to prohibit retraining.

We aim to prevent the language models from generating fabricated information. To this end, we restrict their use to information identified through our bespoke web search process.

However, some tasks inherently require further aggregation of already aggregated information. For example, when estimating the cost shares of system components (see Step 5), we provide excerpts from various sources about the components in question, along with aggregated price and prevalence data for different component types. The model is then instructed to assign a weight between 0 and 1 to each component, such that all weights sum to 1.

This process is not random, but the only trace of the model's reasoning is the summary we record explaining why a particular weight was assigned. These weights can therefore be considered estimates produced by a black-box model based on explicit qualitative evidence. By analogy, the procedure resembles asking a subject-matter expert to assign weights based on the provided information: the results are not fully deterministic but can be expected to fall within a reasonable range.

2.2 Step 0: Select a value chain anchor product.

In the current version of the algorithm, this product can be considered either a material (e.g., stainless steel) or a technical system (e.g., a solar PV system). By technical system, we mean an ensemble of components that together produce effects by design (see the definition of artificial instrumental systems in (Schollmeyer and Tamuzs, 2019)). For pragmatic reasons, we distinguish technical systems from materials by assuming that a technical system can be mechanically disassembled into its functionally distinct parts, whereas materials cannot be separated into such components in any meaningful way by mechanical means alone, but instead require non-mechanical processing to isolate their functionally distinct parts. This distinction is pragmatic rather than ontological; it simply reflects the way these terms are typically used in everyday technical language. We will refer to both technical systems and materials collectively as ***technologies***, since our analysis considers them solely in instrumental terms.

2.3 Step 1: Disambiguate the value chain anchor product (technical system or material) into its most prevalent types.

Before breaking down a system (or material) into its parts, we first need to make sure it is clearly defined. For example, stainless steel exists in several forms. Austenitic stainless steel contains nickel, while ferritic stainless steel may contain little or no nickel.

The same idea applies to solar PV systems, which can be grid-tied, off-grid, or hybrid. These three types share some components, like PV modules, but differ in others; for instance, only hybrid and off-grid systems include battery banks.

To make this distinction clear, we will use ***technology category*** to refer to the general class (e.g., stainless steel, solar PV system), and ***technology type*** for a specific version within that category (e.g., ferritic stainless steel, off-grid solar PV system). This distinction is crucial for the way we construct the technology graph.

Note that the above distinction of solar PV systems into grid-tied, off-grid, and hybrid is logically complete but not the only meaningful way to classify solar PV systems. In fact, searching for types of solar PV systems reveals many alternative classifications that are not based on grid-tiedness but rather on features such as transportability or compactness. Following the traceability principle, our algorithm is instructed to base the distinction of types on what is found in the literature. In cases where conflicting classifications occur, we apply a ranked list of property classes to determine which distinction takes precedence (with color being the least useful category).

Furthermore, we use the web search approach to enrich information about these types in a targeted manner. For example, we search for and extract details on each technology type's function, mechanism, specifications, core components, core distinguishing features, and price distribution (the latter based on reference products found on Alibaba).

2.4 Step 2: Map the technology types to their corresponding 6-digit HS codes.

We map only technology types to HS codes, as we consider these to be sufficiently well-defined products to qualify as tradable goods. Each technology type is assigned to a 6-digit HS code.

Our algorithm follows the General Rules for the Interpretation of the Harmonized System (World Customs Organization, 2025), proceeding from a 2-digit to a 4-digit, and then to a 6-digit pre-selection, before ultimately determining the final 6-digit HS code. The reasoning process is made explicit, with references to the relevant General Rules where appropriate.

2.5 Step 3: Estimate the prevalence of each technology type

To assess the importance of each technology type relative to others, we apply the web search approach described above to estimate the prevalence of each type. This information is then used to assign a weight between 0 and 1 to each type, relative to the system or material it represents. These estimates are generated by a language model based on the available prevalence information for each type. For any given system or material, the prevalence weights of all associated types must sum to 1.

For example, grid-tied solar PV systems have been deployed significantly more frequently than off-grid or hybrid systems. Consequently, their prevalence weight is consistently higher than that of off-grid or hybrid systems. That said, each model run yields slightly different values, since these are aggregated estimates based on other aggregated data. One way to address this uncertainty is to run the estimation process multiple times to bootstrap a confidence interval.

2.6 Step 4: Decompose each technology type into its fundamental parts (i.e., systems and materials)

Once the technology types have been clearly distinguished, each is decomposed into its fundamental parts, which may themselves be systems or materials serving distinct functions.

The decomposition procedure differs depending on whether the technology type represents a system or a material, as the rules governing decomposition vary in each case. For every identified part, we also extract additional information, including its functions, mechanisms, and specifications.

These parts are referred to as technology categories, since they are defined by function, and a given function may be realized in multiple ways through different *technology types*.

2.7 Step 5: Estimate the cost share of each part relative to other parts, labor, additional inputs, and machinery

To assess the relative importance of each part, we estimate its cost share in relation to the other parts, as well as to the estimated costs of labor, additional inputs, and machinery. This information is identified through the web search approach described above.

The cost-share weights for each component and for the additional cost categories (labor, additional inputs, and machinery) represent highly aggregated estimates based on secondary aggregates. A language model produces these estimates using information about each part, available information on labor, machinery, and input costs (as found in the literature), the prevalence of the respective part types, and their price distributions, where identifiable, for example, on Alibaba. The model is instructed to assign a weight between 0 and 1 to each cost category, such that all weights sum to 1.

2.8 Repetition of Steps 1 to 5

For every identified part, we repeat Steps 1 to 5:

1) distinguishing its types,
2) mapping those types to 6-digit HS codes,
3) estimating their prevalence,
4) decomposing each type into its fundamental parts, and
5) estimating the cost shares of each part relative to the corresponding labor, machinery, and additional input costs.

This recursive process is applied five times. In the final iteration, each decomposition is forced to resolve into materials. The decomposition of a branch stops earlier if the identified material type is no longer the result of material processing but can instead be considered a raw material input. As a result, we obtain a graph with paths of the following structure:

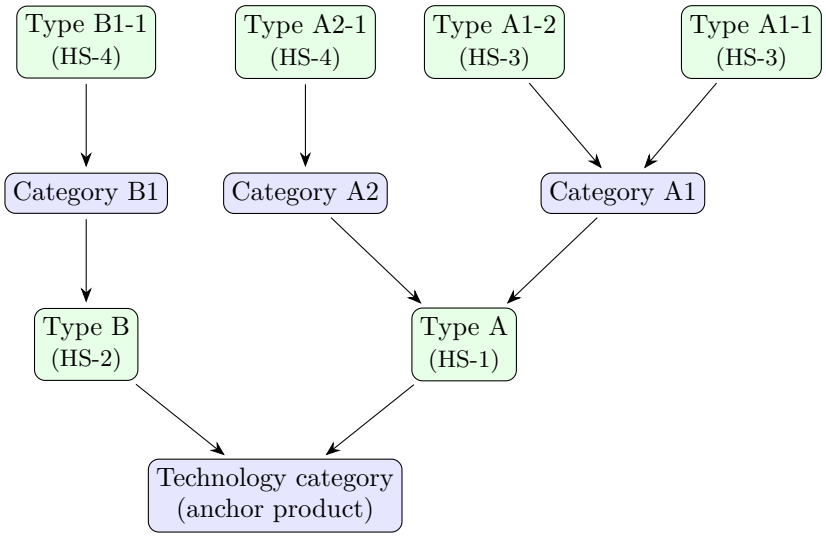

Figure 1: Recursive decomposition: category (anchor product) ← types ← categories ← types (repeated). The technology-type nodes are assigned HS codes. Here, we schematically distinguish between different HS codes using labels such as HS-1, HS-2, etc.

3 Mappings

3.1 Company Mappings

3.1.1 By taxonomy such as NACE or ISIC

To map companies to their position in the value chain, different procedures can be applied depending on data availability and the required level of precision. Business registries provide an almost complete list of existing companies. Depending on the practices of national statistical agencies, these companies are often classified according to their main activities using taxonomies such as NACE or ISIC. By mapping the products in the value chain to these taxonomies – similarly to how they are mapped to 6-digit HS codes – companies can then be linked to a version of the value chain aggregated at the level of detail permitted by the taxonomy.

3.1.2 By Product Offering

The highest-resolution approach to mapping companies to their position in the value chain is through their product offerings. To achieve this mapping for a given region, we conduct targeted, regionally constrained web searches for each item resulting from Step 4 (i.e., the technical systems and materials that constitute specific technology types) to identify companies that sell versions of these items.

This process has at least two limitations. First, we can only identify companies that maintain a web presence. Second, the information on company websites must be taken at face value, meaning

that any assessment of whether a company is a producer, reseller, or service provider must rely solely on that information.

The advantage of this approach is that even small producers can be identified, provided they maintain an active website.

3.2 Talent Mappings

Similar to how we identify companies involved in the production or distribution of products in the value chain through web search, we identify talent by searching for related patents, publications, and research profiles. This approach is useful for regional stakeholder identification and engagement.

3.3 IP Mappings

To identify intellectual property, we follow a similar process. For each element in the value chain, we conduct targeted patent searches to map IP holders to their corresponding positions within the value chain.

4 Value Shares

4.1 Highest resolution (technology categories and technology types)

The value share of an item (*technology type* or *technology category*) in the value chain is the sum of the products of all path weights along the direct routes from the item to the root anchor product.

Formally, for a directed acyclic graph $G = (V, E)$ with normalized edge weights w_{uv}, the value share of item i relative to the root r is defined as:

$$V_i = \sum_{p \in \mathcal{P}(i,r)} \prod_{(u,v) \in p} w_{uv}$$

Where:

$G = (V, E)$	directed acyclic graph representing the value chain, where V is the set of nodes (items) and E is the set of directed edges connecting them
V_i	value share of item i in the value chain
r	the root (a unique node representing the anchor product)
$\mathcal{P}(i, r)$	set of all direct acyclic paths from item i to root r
(u, v)	a directed edge connecting two nodes along a path p, such that $(u, v) \in E$
w_{uv}	normalized edge weight for (u, v): either prevalence (type→category) or cost share (category→type), with outgoing edge weights from each node summing to 1
$\prod_{(u,v) \in p} w_{uv}$	product of weights along path p
$\sum_{p \in \mathcal{P}(i,r)}$	sum across all directed acyclic paths from i to r

4.2 First aggregation (technology types only)

On the graph's highest resolution level, it does not matter whether we are looking at technology types or technology categories. However, if we want to focus only on technology types – that is, concrete products that have been mapped to HS codes – we need to "jump" over the technology categories (see Figure 1) and convert the graph into the more aggregated form:

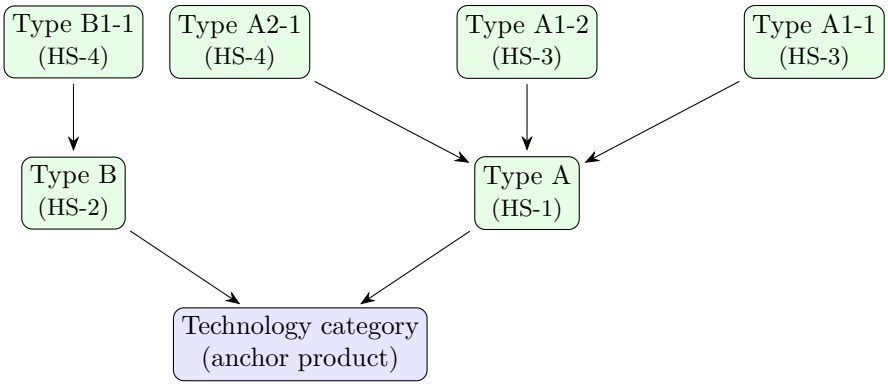

Figure 2: Flattened types-only graph: category (anchor product) ← types ← types (repeated). Each technology-type has assigned an HS code. We schematically distinguish between different HS codes using the same labels as in Figure 1.

In this flattened representation, every two-step connection (technology type → technology category → technology type) is replaced by a single composite edge that directly connects one technology type to another.

4.2.1 Composite edge weight

Each composite edge weight W_{ab} is defined as the sum of all possible two-step connections through intermediate technology categories $\mathcal{C}$ linking technology type T_a to technology type T_b:

$$W_{ab} = \sum_{C \in \mathcal{C}(a,b)} \pi_{aC}\, \sigma_{Cb}$$

Where:

π_{aC}	the prevalence weight, representing how commonly technology type T_a occurs in category C
σ_{Cb}	the cost-share weight, representing the relative cost contribution of C in constructing T_b
$\mathcal{C}(a,b)$	the set of all technology categories that connect T_a and T_b

4.2.2 Value share of a technology type

Because the root anchor product r is a technology category, we introduce a terminal weight t to represent the connection from a technology type t to the root. The value share of a technology type T_i is then calculated as the sum of the products of all composite weights along every direct path leading from T_i to the root anchor product, multiplied by the terminal weight of the final node in the path:

$$V_i = \sum_{p \in \mathcal{P}(T_i, r)} \left(\prod_{(u,v) \in p} W_{uv} \right) \rho_{\text{terminal}(p)}$$

Where:

$\mathcal{P}(T_i, r)$	the set of all paths from technology type T_i to the root anchor product r
W_{uv}	the composite prevalence–cost-share weight between nodes u and v
$\rho_{\text{terminal}(p)}$	the terminal weight connecting the last technology type in path p to the root anchor product

4.3 Second aggregation (HS Codes)

Since every technology type has an associated 6-digit HS code, we can aggregate the technology-type graph (Figure 2) into an HS code graph (Figure 3). In this representation, each HS code node encompasses one or more technology types that share the same classification.

Conceptually, this second aggregation layer replaces all individual technology types within an HS code with a single node representing that code. The resulting graph maintains the same hierarchical structure as before, but now the nodes represent HS codes instead of individual technology types:

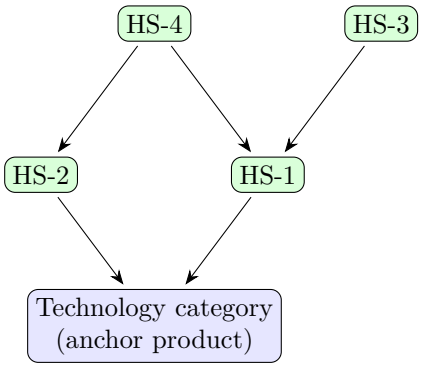

Figure 3: HS code aggregation graph: category (anchor product) ← HS codes ← HS codes (repeated). The schematic HS codes are the same as in Figure 1 and Figure 2.

4.3.1 Composite edge weights between HS codes

Each edge between two HS codes H_a and H_b is defined by aggregating over all possible connections between the technology types associated with these codes. Let $T(H_a)$ denote the set of technology types belonging to HS code H_a and similarly $T(H_b)$ for HS code H_b.

The composite HS-to-HS weight W_{ab}^{HS} is given by:

$$W_{ab}^{\text{HS}} = \sum_{i \in T(H_a)} \sum_{j \in T(H_b)} W_{ij}$$

where W_{ij} are the type-to-type composite weights defined in the previous section. This formulation ensures that every indirect connection between the technology types grouped under H_a and H_b contributes to the aggregated HS-level relationship.

4.3.2 Value share at the HS code level

Analogous to the technology-type graph, the value share of an HS code H_i is defined as the sum of the products of HS-level composite weights along all direct paths leading from H_i to the root anchor product:

$$V_i^{\text{HS}} = \sum_{p \in \mathcal{P}(H_i, r)} \left(\prod_{(u,v) \in p} W_{uv}^{\text{HS}} \right) \rho_{\text{terminal}(p)}$$

Where:

$\mathcal{P}(H_i, r)$	the set of all paths from HS code H_i to the root anchor product r
W_{uv}^{HS}	the aggregated HS-to-HS edge weights
$\rho_{\text{terminal}(p)}$	the terminal weight connecting the last HS code in the path to the root category

This second aggregation produces a trade-level representation of the value chain that is directly compatible with international trade data.

By collapsing multiple technology types into HS codes, we can analyze and compare value-chain structures across countries using trade data, while still preserving the internal logic of the original technology-type graph.

5 Limitations

5.1 Data and Source Limitations

The method inherits several biases and uncertainties from the uneven distribution of technical and market information across domains. The identification of necessary parts of technical systems, their technology types, relative prevalence, and component-level cost shares depends heavily on the availability of reports, patents, articles, and web documentation. The presence and quality of such information vary widely between technologies. Where explicit evidence is missing, estimates cannot be inferred from traceable source texts but are potentially entirely generated by the language model itslef, introducing additional uncertainty.

A further source of bias arises from the reliance on a single commercial marketplace (currently Alibaba) for price references. This concentrates evidence in one geography and supplier type, skewing the estimated price distributions toward visible, export-oriented producers. Firm-level mapping suffers from a similar coverage limitation: companies without a digital footprint, or with ambiguous website descriptions, are underrepresented or misclassified, reducing representativeness in the reconstructed production landscape.

Marketplace price data, while useful as one of the inputs for first approximations, also risks conflating prices with costs. Listed prices embed markups, logistics costs, and policy distortions that make them poor proxies for actual production costs. Finally, both price and prevalence information are time-sensitive. Without explicit timestamps and regular re-sampling, the resulting technology graphs may lag behind real-world industrial dynamics and fail to reflect recent innovation or cost evolution.

5.2 Structural and Taxonomic Limitations

The recursive decomposition is currently capped at five levels, a pragmatic constraint to control the exponential growth of the knowledge graph. However, this fixed depth may truncate deeper or more complex production hierarchies.

A related issue concerns the ambiguity of technology typing. Technology types can legitimately be distinguished along multiple orthogonal dimensions, such as functional principle, material composition, design architecture, or intended application. No single rule ensures that one dimension prevails across all cases, making consistent disambiguation difficult. Moreover, the boundary between categories and types is itself fluid: in practice, the model must judge how specific categories should be before type distinctions appear. Our experiences show that both the choice of model architecture and the number of textual sources influence how this partitioning is carried out, leading to possible variability across runs.

Additional structural uncertainty arises from divergent classification systems. Literature, engineering practice, and trade nomenclature often impose conflicting hierarchies. Even when the Harmonized System (HS) mapping follows the General Rules for Interpretation (GRI), many-to-one and borderline cases at the 6-digit level inject noise and inconsistency, partly reflecting ambiguities already present in the trade data itself.

5.3 Methodological and Epistemic Limitations

At the methodological level, the workflow remains partly stochastic. The language model's internal weighting of textual evidence means that both prevalence and cost-share estimates can vary between runs. These differences reflect a combination of random sampling, prompt sensitivity, and retrieval variation. One way to address this limitation is through a bootstrap procedure, running the same query multiple times to construct empirical confidence intervals for each estimated weight.

Black-box behavior also extends to higher-order decisions such as the generation of technology categories and type distinctions. These outputs depend strongly on model architecture, request limits for textual inputs, and specific prompt formulations, all of which may shape the inferred technological taxonomy.

Ensuring full traceability across recursive model runs adds another layer of difficulty. Maintaining a provenance ledger that captures source URLs, timestamps, extraction snippets, and model versions is computationally and architecturally intensive, particularly when applied to large-scale reconstructions.

Finally, the model aspires to approximate a global representation of technological organization, but this ideal is inherently limited. Cost structures and technology types vary regionally due to path dependencies, industrial specialization, and regulatory environments. Any global weighting scheme therefore smooths over real geographic heterogeneity, and the resulting system should be interpreted as an approximate abstraction rather than a precise empirical map.

6 Future Work

…

References

Schollmeyer, J., Tamuzs, V. 2019. Deducing altshuller's laws of evolution of technical systems. International TRIZ future conference, Springer, 55–69.

World Customs Organization. 2025. HS rules. https://www.wcotradetools.org/en/harmonized-system/rules.