**GraphRAG-Enhanced Information Extraction for Sustainable Masonry Decision Support**

Dominik Stellmacher1 | Karina Große Lögten2| Agnes Kelm1 | Anica Meins-Becker1

**Correspondence**

Dr. Paul Smith

University of Wuppertal

Institute for Net Zero Studies

Gaußstreet 500

42119 Wuppertal  
Email: [p.smith@uni-wuppertal.de](mailto:p.smith@uni-wuppertal.de)

1 Department of Digital Design, Construction and Operation, University of Wuppertal, Germany

2 List Eco, Cologne, Germany

**Abstract**

Reliable, standards-compliant data are essential for sustainable material decisions in the construction industry; however, such information is often only available in unstructured PDF formats, such as technical, safety, or Environmental Product Declarations (EPDs). This study presents an end-to-end pipeline based on a graph-based Retrieval-Augmented Generation architecture (GraphRAG). The pipeline converts EPDs into standardized JSON structures using rule-based processing, while other document types are transformed into TXT or CSV/Excel formats. Products are classified in compliance with relevant standards—by trade, product group, and type—using embeddings with large language model (LLM)-assisted validation. Variant-specific, standards-relevant attributes are extracted with harmonized units, and all results, including source and quality metadata, are stored in a knowledge graph. Applied to ten masonry products, the system achieved 100% type classification accuracy (70% without validation) and precision = recall = 1.00 for variant detection; all metrics passed self-consistency validation. The graph-based representation enables multi-criteria queries—such as selecting the most ecologically favorable product for given performance requirements—and provides an auditable foundation for multi-criteria decision-making and assessment frameworks.

**Keywords**

Product Selection, Decision Support System, Large Language Model, Self-Consistency Validation, Variant Detection

# Introduction

The construction sector accounts for an estimated 37% of global energy- and process-related CO₂ emissions [1]. Achieving climate policy goals, such as those of the European Green Deal, requires that construction product selection systematically consider not only economic and technical, but also ecological and socio-cultural criteria [2]. Multi-Criteria Decision Making (MCDM) provides a structured framework for this, enabling product choices to be assessed using environmental indicators, costs, and performance- and health-related data [3]. For sustainable construction products, a data-driven MCDM process is essential, as only the integrated evaluation of all relevant factors ensures robust decisions.

In practice, much of the required product information exists in unstructured PDFs—such as technical data sheets (TDS), safety data sheets (SDS), Environmental Product Declarations (EPD), or Declarations of Performance (DoP). Over 85% of building-related data is unstructured [4], forcing practitioners into time-consuming manual searches and leading to frequent information loss. Despite the abundance of data, about 96% of collected project information remains unused [5] due to the lack of systematic processing—creating a “dark data” barrier to data-driven decision-making.

This paper addresses this gap with an approach for automated information extraction from unstructured construction product documents. Within the research project NaConBau (Sustainable Controlling Tool for Construction Contractors, funded by the German Federal Ministry for Economic Affairs and Climate Action) [link], a pipeline was developed that extends the Retrieval-Augmented Generation (RAG) concept with knowledge graph integration—GraphRAG. Building on the RAG framework by Lewis et al. [6], which combines a language model with external memory for fact-based text generation, GraphRAG explicitly incorporates a knowledge graph as an external knowledge base [7]. While GPT-based LLMs are already in productive use in the construction sector, they show methodological limitations, underscoring the need for retrieval and knowledge graph enhancements [8]. GraphRAG enables not only the retrieval of document passages but also the exploitation of relational links between product data, supporting complex queries and multi-criteria analyses.

The objective of this work is to analyze the core components of the GraphRAG pipeline—particularly product type classification and standards-guided information extraction—from a scientific perspective. Specifically, the study investigates:

1. How construction products can be automatically assigned to a product type based on their documentation.
2. Which product type-specific attributes are relevant for decision-making and need to be extracted.
3. How these attributes can be accurately and consistently retrieved from the documents.

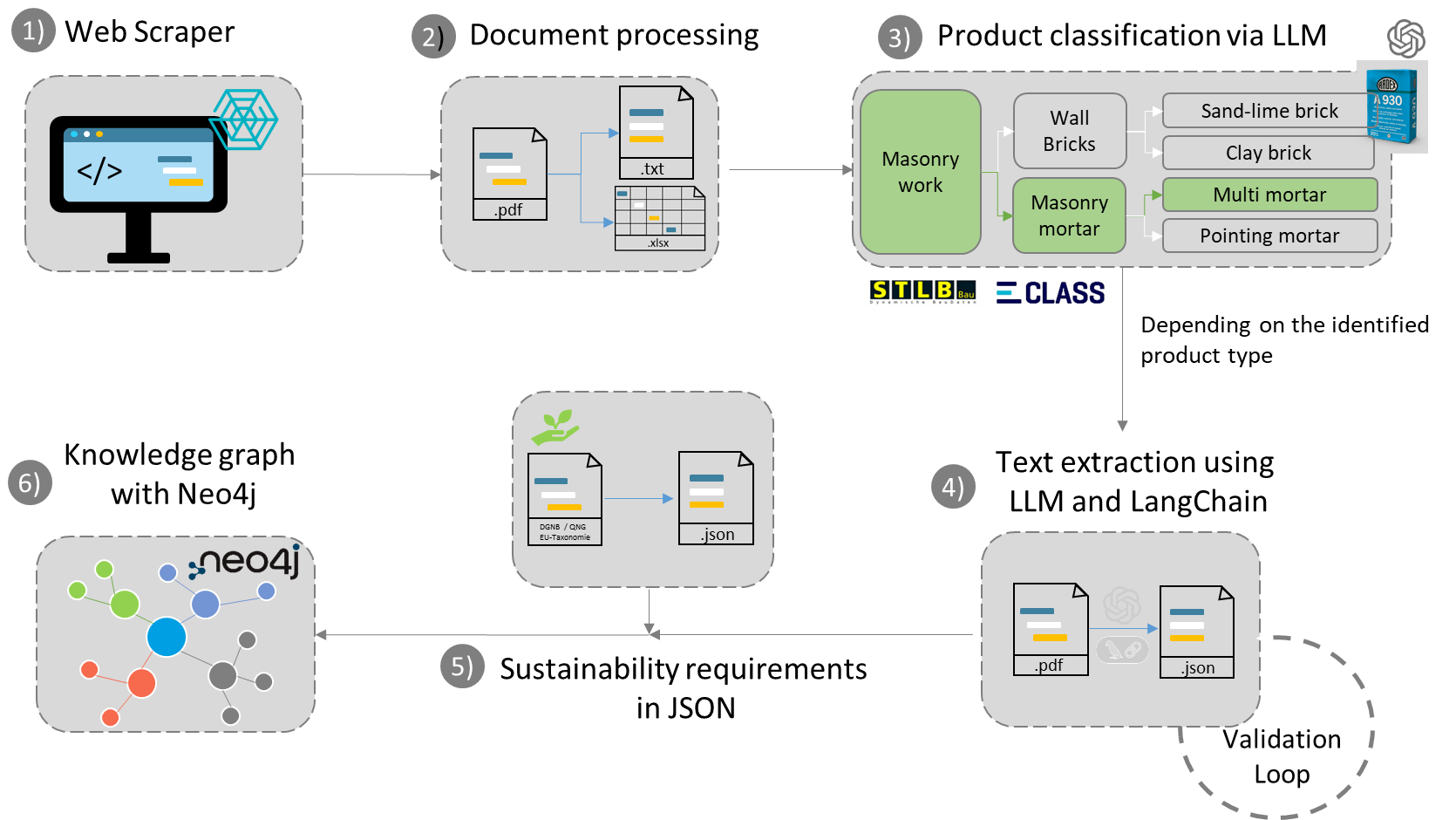
The subsequent sections outline the methodology (Section 2) and present the experimental results (Section 3), followed by conclusions and an outlook (Section 4).

# Methodik

The developed pipeline encompasses—illustrated in Figure 1—the complete process from document retrieval to knowledge provision.

Initially, (1) unstructured PDF documents of the considered construction products are directly crawled from manufacturer websites. In step (2), these are converted into structured, machine-readable formats (e.g., TXT and CSV) to remove distracting layout elements and reduce the token count for LLM queries. In step (3), the product is assigned to a predefined product type—such as sand-lime brick or general-purpose masonry mortar—based on its associated datasheets. Step (4) then extracts product-type-specific relevant attributes, considering different product variants. In step (5), the results are compared against the sustainability requirements of various certification and assessment systems (e.g., EU Taxonomy, DGNB, QNG), before step (6) stores all product data together with requirements in a graph database. Finally, step (7) uses this knowledge base to power a decision support system that can filter and evaluate sustainable construction products according to freely combinable criteria via graph queries. The graph architecture enables semantically rich decision support, as relationships such as *“Product A has compressive strength B”* are explicitly navigable in the knowledge graph. Overall, the pipeline realizes an end-to-end workflow from the ingestion of unstructured data through AI-assisted interpretation to persistence in a queryable knowledge graph.

The following subsections focus specifically on steps 2–4, as they are central to automated information preparation and extraction.



**Figure 1** Processing steps of the GraphRAG pipeline

## Document Preprocessing (Step 2)

In the preprocessing stage, heterogeneous PDF documents (technical data sheets, safety data sheets, environmental product declarations, declarations of performance) are transformed into a structured, analyzable form. Running text is extracted as plain text, and tabular content is converted into CSV/Excel. This approach eliminates the visual layout noise of PDFs and reduces the token count for subsequent LLM queries, improving efficiency and extending the available context window.

For Environmental Product Declarations (EPDs), much of the processing can be rule-based, as they follow harmonized standards in construction (DIN EN 15804). Many EPD programs—such as the *Institut Bauen und Umwelt e.V.* (IBU)—use standardized templates. Using predefined multilingual key phrases (e.g., *“RESULTS OF THE LCA”*), structural markers (table headers such as *Name – Value – Unit*), and controlled synonym lists, the preprocessing step segments the EPD into logically distinct sections. Irrelevant layout elements are removed, and contents are normalized.

The result is a set of well-defined tables, including:

1. Declaration header (manufacturer, declaration number, validity period),
2. Declared unit,
3. Service life and product lifespan,
4. Environmental impact indicators (e.g., GWP, AP, EP, POCP as per EN 15804), and
5. Resource use indicators (e.g., renewable/non-renewable primary energy—PERE/PENRE, recycled content, waste quantities).

These are stored together with their life cycle phase allocations (Modules A1–A3, B-phases, C, D in accordance with DIN EN 15978).

All partial EPD information is then transformed into a unified JSON schema. The structured EPD JSON contains flat key–value fields for header, unit, and service life entries, along with a phase-resolved list of environmental and resource data using canonical indicator names. For certain mandatory attributes from standards or assessment systems (e.g., TVOC/SVOC content, R-value, VOC without NIK, proportions of carcinogenic substances, AgBB compliance), a two-stage approach is used: regular expressions first scan the full text; any remaining gaps are filled through targeted LLM queries.

This produces a consistent, normalized, and phase-specific dataset that can serve directly—without manual intermediate steps—as a reliable input for classification, extraction, and graph persistence. Explicitly leveraging the standardized EPD structure—particularly for IBU EPDs—results in high extraction quality and reproducibility, while LLM fallbacks reliably complement occasional missing information (e.g., AgBB *VOC without NIK* specification).

## Product Classification and Assignment (Step 3)

### Product Classification

To ensure consistent extraction and evaluation, each construction product must be uniquely assigned to a product type. The developed classification component operates in three stages:

1. Determination of the trade according to the Standardleistungsbuch Bau (STLB-Bau) [9],
2. Assignment to a product group within the trade, and
3. Subdivision into the specific product type.

For example, a product in the masonry trade is assigned—based on applicable standards (DIN EN 771-1 to -6 for masonry units, DIN EN 998-2 for mortar, DIN EN 845 for ancillary components)—to one of the following groups: “load-bearing masonry units,” “non-load-bearing masonry units,” “masonry mortar,” or “ancillary products.” Within “load-bearing masonry units,” type differentiation follows, e.g., clay brick, sand-lime brick, lightweight concrete block, or autoclaved aerated concrete block.

Products that cannot be assigned to a single trade (e.g., insulation materials or adhesives) are treated as overarching construction products: in these cases, the trade level is omitted in favor of a general category, and stage (ii) starts directly at the product group (e.g., insulation, adhesives), from which concrete product types (e.g., between-rafter insulation, dispersion adhesive) are derived.

### ****Product Type Assignment****

The automatic assignment of an unknown product document to a specific product type is two-stage and combines:

1. semantic embedding-based preselection and
2. LLM-assisted validation.

**(a) Embedding-based preselection:** For each possible product type, a normative, attribute-based definition serves as a semantic anchor (see Listing 1 for sand-lime brick). Both these definition texts and the consolidated document content from the PDB-TXT files are vectorized using the OpenAI text-embedding-3-small model. Cosine similarity is then computed between all type anchors and the document vector. The type with the highest similarity is passed as the sole candidate (suggested\_type) to the next stage (no top-k list is generated). Explicit type definitions reduce ambiguities (e.g., sand lime vs. concrete block) and stabilize preselection despite manufacturer-specific product names.

|  |
| --- |
| *“A masonry unit standardized according to DIN EN 771-2, made from slaked lime and sand, with nominal dimensions, e.g., 490 × 240 × 238 mm (± 2 mm). Bulk density 1,900–2,300 kg/m³, compressive strength class CS 12–CS 50 (12–50 N/mm²), water absorption ≤ 12 wt%, thermal conductivity λ 0.7–1.3 W/(mK). Frost and de-icing salt resistant (XF1–XF4), non-combustible (A1), used for load-bearing and non-load-bearing masonry as well as acoustically and thermally optimized walls.”* |

Listing 1 Example of a product type definition (sand lime brick)

**(b) LLM Validation: The consolidated document, the embedding suggestion, and the complete list of type definitions are provided to a GPT-4 model. A structured system prompt instructs the model to examine, step-by-step:**

1. whether a product type name is explicitly mentioned in the text,
2. which material attributes match the definition criteria, and
3. which type overall fits best.

**It must then output a formalized two-line response:**

|  |
| --- |
| *Confidence: <Number> # self-assessed 0–100*  *<ProductType> # only the type name* |

Listing 2 Specification of formalized output

The detailed chain-of-thought is logged internally (if enabled) but is not written into the JSON results. After validation, the confirmed product type is uniquely mapped to the product group and trade category (e.g., sand-lime brick → masonry units → masonry work). The system then loads the corresponding extraction module (e.g., SandLimeBrick.py or ThinBedMortar.py), which implements the standards-based verification and parsing logic for technical, ecological, and socio-cultural attributes.

|  |
| --- |
| *You are an expert assistant for construction materials and help assign a construction material document to a product type. You receive content from multiple documents containing relevant information about a construction material. Think extensively in the background (chain-of-thought) and follow these steps:*   1. *Carefully read the combined contents from all documents and identify relevant keywords and topics.* 2. *Focus on those that indicate material properties, application areas, or product descriptions.* 3. *Check whether the name of a product type from the product type list is explicitly mentioned in the text. If so, the product should with high probability be assigned to that product type.* 4. *Compare this content with the descriptions of the product types in the list and analyze which product type fits best.* 5. *For each product type, explain why it fits or does not fit, based on specific characteristics found in the combined documents.* 6. *Select the product type that fits best.* 7. *Estimate your classification confidence as an integer between 0 and 100.*   *List of product types:*   * *Sand-lime brick: …* * *Clay brick: … ...*   *At the end, output exactly two lines:*  *Confidence: <Number>*  *<ProductType>* |

Listing 3 Excerpt from system prompt for CoT-based product type validation

Listing 3 shows the system prompt for CoT-based product type validation. By coupling an explicit, standards-based definition corpus with statistical semantics (embeddings) and rule-driven LLM validation, the method achieves high type specificity. Only when a product is clearly assigned to the correct product type can the subsequent extraction and evaluation processes be carried out reproducibly and with precision.

## ****Variant Detection and Information Extraction (Step 4)****

Once the product type is determined, the corresponding product-type-specific extraction module (e.g., SandLimeBrick.py or NormalMasonryMortar.py) is executed. Extraction proceeds in a three-stage pipeline:

1. variant detection,
2. information extraction per variant, and
3. validation/quality assurance.

**(a) Variant Detection:** In some technical data sheets, multiple product variants are described together—e.g., different brick dimensions or various packaging forms of mortar. Since such variants may differ in performance-critical properties (a larger brick may have higher sound insulation), each variant described in the document must first be identified.

Detection is performed by an LLM analyzing the document text extracted from PDB-TXT files. A dedicated system prompt instructs the model to output all variants as a JSON array; each element contains at least "variant\_name" and an "dimensions" field L,W,HL, W, HL,W,H, optionally extended with "stone\_type", "compressive\_strength\_class", "bulk\_density\_class", and other features (see Listing 4 for prompt structure). If no variants are detectable, a default variant is generated automatically.

|  |
| --- |
| *"You are an assistant that identifies all distinct product variants in a technical data sheet (PDB\_) and extracts relevant information for each variant. A variant may differ by dimensions (e.g., L×W×H), stone types (e.g., solid unit, hollow unit), compressive strength classes, bulk density classes, or other distinguishing characteristics. Create a JSON array (without code fences) containing all detected variants. Each variant is an object:\n" " "variant\_name": "...", \n "dimensions": [L, W, H], \n "stone\_type": "solid"/"hollow" or empty, \n "compressive\_strength\_class": "..." or empty,\n "bulk\_density\_class": "..." or empty,\n "other\_features": "..." (if you find any)\n "}\n"*  *"If no information on stone type, strength class, etc. exists, leave those fields empty. Return only a pure JSON array, with no explanations. Think step by step (chain-of-thought) and analyze the document text. Then summarize the detected variants in a JSON array.\n\n" )* |

Listing 4 System prompt for variant identification

The resulting structured variant list drives subsequent extraction: for each variant, target fields—such as compressive strength or thermal conductivity—are again queried via LLM, with the respective dimension provided as context. Each extracted field undergoes a self-consistency loop with multiple queries and majority-vote consensus building (validate\_extracted\_field). Units are harmonized (e.g., MPa ↔ N/mm², kg/dm³ → kg/m³), and numerical values are converted to standard units to enable cross-manufacturer comparisons. The output includes the identified source, the raw value with original unit, the converted comparison value, and a confidence score.

**(b) Information Extraction per Product Type:** The attributes to be extracted vary by product type. Normative foundations are the European product standards for masonry products (DIN EN 771-1 to -6) and DIN EN 998-2 for masonry mortar. A catalog of mandatory and optional attributes was derived, defining field names, standard units, and a specific query prompt for each attribute. (Economic data are not considered, as technical data sheets typically do not contain pricing information.)

Minimum required attributes—common to all masonry unit types (per DIN EN 771)—include:

* **General information:** product name, manufacturer, product description (short characteristics, e.g., material), field of application (load-bearing/non-load-bearing)
* **Technical properties:** dimensions and permissible tolerances [mm]; shape and hole pattern; bulk density (dry, gross/net) [kg/m³]; compressive strength (single unit, fb) [N/mm²]; bond strength (shear bond or flexural bond tensile strength) [N/mm²]; water absorption (capillary) [kg/(m²⋅s0.5)] or [wt %]; water vapor diffusion resistance μ; thermal conductivity λ₁₀,dry [W/(mK)]; durability (frost/de-icing salt resistance, class F1/F2); reaction to fire (Euroclass A1–F); hazardous substances (declaration per normative chapter incl. national requirements)
* **Ecological property:** availability of an EPD (yes/no)
* **Socio-cultural property:** information on hazardous ingredients (e.g., from safety data sheets)

Product-Type-Specific Additional Attributes: In addition to the common set of core parameters, each masonry unit type requires the declaration of specific supplementary characteristics. For example, clay masonry units (EN 771-1) must report the soluble salt content (classes S0–S2), the initial rate of water absorption, and the typical moisture movement. Calcium silicate masonry units (EN 771-2) are required to declare the typical moisture movement (shrinkage). Concrete masonry units (EN 771-3) must provide flexural strength or flexural bond strength, as well as moisture movement. Autoclaved aerated concrete masonry units (EN 771-4) must state drying shrinkage and swelling, as shrinkage behavior is critical for their long-term performance. Manufactured stone masonry units (EN 771-5) additionally require flexural bond strength and typical moisture movement, but not open porosity or flexural strength. These latter two parameters—open porosity and flexural strength—are normatively required only for natural stone masonry units (EN 771-6), which must also declare flexural bond strength and, where applicable, a petrographic description.

For masonry mortar (EN 998-2), parameters such as mortar group or class, shear bond strength, water vapor diffusion resistance, workability time, and, where applicable, frost resistance (suitability for specific exposure classes) must be declared. These normative specifications are embedded directly into the extraction logic of the implemented modules. The process operates fully automatically, using predefined query prompts uniquely assigned to each target field. Following the preparation of a consolidated text-and-table corpus—prioritized as PDB-Text, then PDB-PDF, and finally PDB-Excel—the module iterates systematically over all target fields. Each prompt specifies the product type, provides a standard- and unit-specific description of the desired property, and instructs the system to return only the relevant value or a {value, unit} JSON structure.

Variant-dependent parameters (e.g., a specific width) are dynamically inserted into the prompt. Model responses are parsed, units are converted into standardized formats (e.g., kg/dm³ → kg/m³, MPa ↔ N/mm²), and numerical values are normalized to ensure that all manufacturer-provided specifications are directly comparable. To avoid semantic ambiguities, the prompts are precisely aligned with the underlying standard’s logic; Listing 5 illustrates this with the fields compressive strength fb and compressive strength class, together with their field-specific query specifications.

|  |
| --- |
| ***Compressive Strength fb***  *“Extract only the declared single-unit compressive strength f₍b₎ of the calcium silicate brick in accordance with DIN EN 771-2 / DIN 20000-402. Return a value only if f₍b₎ is explicitly stated in the document as a numeric value. Do not use the characteristic masonry compressive strength f₍k₎ and do not use the compressive strength class. If the document contains only a compressive strength class (e.g., 12), return None (no derivation). Output: numeric value in N/mm² (MPa = N/mm²; convert if necessary), with a maximum of two decimal places, without unit. Examples: 12 or 12.50.”*  ***Compressive Strength Class***  *“Extract only the compressive strength class of the calcium silicate brick in accordance with DIN EN 771-2 / DIN 20000-402 (e.g., 12, 16, 20). Do not extract the single-unit compressive strength f₍b₎ and do not extract the characteristic masonry compressive strength f₍k₎. Do not return mortar classes (DIN EN 998-2), bulk density classes, or strength groups. Output: pure number (e.g., 12), without unit and without any letter prefix.”* |

Listing 5 Field-specific prompts for compressive strength and compressive strength class

**(c) Validation and Quality Assurance: The** extraction pipeline employs a self-consistency validation combined with a majority-voting approach to ensure the reliability of values returned by the LLM. This method—first systematically described by Wang et al. (2022) [10]—has been shown to measurably improve result robustness. For each target field (including the examples in Listing 5), the same query prompt is executed five times, each with a slightly varied model initialization (temperature = 0.4). The resulting responses are then normalized (e.g., MPa ↔ N/mm², kg/dm³ → kg/m³) and aggregated into candidate clusters whose numerical values deviate by no more than ± 1%. The most frequent value is then selected via majority vote; for numerical fields, the mean value within the dominant cluster is calculated to compensate for minimal deviations. The confidence score is defined as the proportion of matching responses across all runs (e.g., 5/5 = 1.0; 4/5 = 0.8) and is stored alongside the final result for traceability. An automated threshold check (e.g., score ≥ 0.8) is not applied in the current implementation; instead, the score serves exclusively as a transparency measure and for subsequent quality assessment. This quality assurance concept—combining self-consistency sampling with ensemble methods—reflects the state of the art and ensures that even the most critical attributes (see Listing 5) are extracted consistently and reproducibly [11].

# ****Results****

To evaluate the pipeline, ten different construction products from the masonry domain (bricks and mortar from seven manufacturers) were selected. Depending on availability, each product was associated with various document types (technical data sheets, safety data sheets, environmental product declarations, declarations of performance) to test the pipeline under realistic conditions. The following subsections summarize the results of product type classification (3.1), variant detection (3.2), and information extraction (3.3).

## ****Product Type Classification****

Table 1 presents the classification results for the ten test products. Listed are the actual product type according to the manufacturer, the available document types, the product type suggested by the embedding method (with cosine similarity), and the final classification result from the LLM validation (including the model’s reported confidence in %).

Table 1 Type Accuracy of Classification for 10 Construction Products (Embedding vs. CoT) and Available Documents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product Nr. | |  | | --- | | Actual Product Type | | Available Documents | Embedding Suggestion (Similarity) | Final CoT Classification (Confidence) |
| 1 | Sand-lime brick (load-bearing) | TDS, SDS, EPD | Sand-lime brick (0.7093) | Sand-lime brick (95 %) |
| 2 | Sand-lime brick (load-bearing) | TDS, EPD | Sand-lime brick (0.6762) | Sand-lime brick (90 %) |
| 3 | Clay brick (load-bearing) | TDS, EPD | Clay brick (0.6986) | Clay brick (90 %) |
| 4 | Clay brick (load-bearing) | TDS | Autoclaved aerated concrete block (0.7173) | Clay brick (90 %) |
| 5 | Concrete block (load-bearing) | TDS | Sand-lime brick (0.7173) | Concrete block (90 %) |
| 6 | Facing clinker brick (non-load-bearing) | TDS | Facing clinker brick (0.6720) | Facing clinker brick (85 %) |
| 7 | General-purpose masonry mortar | TDS, DSD, DoP | General-purpose masonry mortar (0.6810) | General-purpose masonry mortar (95 %) |
| 8 | General-purpose masonry mortar | TDS, SDS, EPD, DoP | Sand-lime brick (0.6666) | General-purpose masonry mortar (95 %) |
| 9 | Lightweight masonry mortar | TDS, SDS, EPD, DoP | Thin-bed mortar (0.7130) | Lightweight masonry mortar (94 %) |
| 10 | Thin-bed mortar | TDS, SDS, EPD, DoP | Thin-bed mortar (0.6872) | Thin-bed mortar (95 %) |

The pure vector embedding assignment produced an incorrect product type in 3 out of 10 cases (italicized in Table 1). Misclassifications occurred particularly between similar product types—for example, Product 5, a concrete block, was incorrectly positioned closer to calcium silicate brick by the embedding method. However, the subsequent CoT validation by the LLM detected all misclassifications and corrected them. As a result, the correct product type was ultimately identified for all products (100% accuracy). The confidence values output by the LLM (85–95%) reflect the difficulty of certain cases—for instance, Product 5, where the model identified the type concrete block with 90% confidence, even though the embedding initially suggested calcium silicate brick. Overall, these findings show that combining embedding-based preselection with LLM validation yields robust classification results. The table also highlights which document types were available for each product—for example, some mortar products (Nos. 8–10) also had EPDs, which were used for extraction (Section 3.3) but were less relevant for type classification.

## Variant Detection

All ten test products were further analyzed for the presence of multiple product variants. The variant detection stage achieved perfect precision and recall, with no false positives (non-existent variants incorrectly detected) and no false negatives (actual variants overlooked). In total, all 50 of 50 variants were correctly identified, corresponding to Precision = 1.00 and Recall = 1.00. For example, Product 1 (calcium silicate brick) contained seven variants in its technical data sheet, and the pipeline detected exactly those seven. Product 2 listed 28 dimensional variants, all of which were found. Mortar product 8 was available in two delivery formats—bagged and silo—both of which were identified. Products 3, 5, 7, 9, and 10 each represented single-variant cases, and the system correspondingly reported exactly one variant for each.

Table 2 presents an excerpt of the extracted properties for the seven variants of Product 1 (calcium silicate plan brick). As expected, variant-specific parameters—such as dimensions, brick format, and sound insulation index Rw—vary across the columns, whereas cross-variant attributes, including bulk density class and compressive strength class, remain constant. Manual verification confirmed that all values were captured fully and accurately.

Table 2 Extraction of Relevant Properties for the 7 Variants (V1–V7) of Product (Sand-lime brick)

| Property | V1 | V2 | V3 | V4 | V5 | V6 | V7 | Evaluation |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dimensions | 248 × 115 × 248 mm | 498 × 115 × 248 mm | 248 × 150 × 248 mm | 248 × 175 × 248 mm | 298 × 175 × 248 mm | 248 × 240 × 248 mm | 248 × 300 × 248 mm | Fully correct |
| Format | 4 DF | 8 DF | 5 DF | 6 DF | 7,5 DF | 8 DF | 10 DF | Fully correct |
| Bulk density class | 1,8 – identical for all variants – | | | | | | | Fully correct |
| Compressive strength class | 12 – identical for all variants – | | | | | | | Fully correct |
| Weighted sound insulation index Rw) | 49.9 dB | 49.9 dB | 53.2 dB | 55.1 dB | 55.1 dB | 59.1 dB | 62.0 dB | Fully correct |

## Information Extraction and Validation

For each product, all target fields defined in Section 2.3 were extracted and manually cross-checked against the original documents. In most cases, the pipeline accurately captured all intended attributes. However, not every theoretically defined field was present in every data sheet; some manufacturers omit certain recommended entries or phrase them indirectly. Likewise, some documents contained additional practice-relevant data beyond the normative requirements (e.g., a weighted sound insulation index Rw or a characteristic masonry compressive strength f₍k₎ for a specific brick–mortar combination). The original extraction list was therefore iteratively expanded, and the prompts were refined to ensure consistent capture of such additional information. In some cases, prompt definitions had to be adapted to the actual wording used in the documentation—for example, in mortar durability, where instead of a frost resistance class the phrase “suitable for moderately aggressive environments” was used. Two examples illustrate extraction performance:

**Product 1 (Sand-lime brick):** The technical data sheet included all relevant information—compressive strength class, bulk density class, thermal conductivity, fire resistance class, variant-specific dimensions, and sound insulation indices—as well as a declaration of hazardous ingredients in the SDS (high quartz content). The pipeline extracted all fields without error. Table 3 presents an example of the results for Variant 1 (dimensions 248 × 115 × 248 mm), showing the value stated in the document, the extracted value, and the confidence score (CS) from the self-consistency validation. All extracted values matched the manufacturer’s specifications exactly. Unit harmonization functioned as intended: the compressive strength class was returned as the pure number “12” (rather than “12 N/mm²” as in the document), while for thermal conductivity, the value and unit remained unchanged (already expressed in the standard unit W/(mK)). The confidence score was 1.0 for all fields, meaning that identical results were obtained across all five queries. This case represents the ideal scenario of complete and consistent extraction.

Table 3 Extraction Quality for Product 1 – Variant 1 (Calcium Silicate Brick, Format 248 × 115 × 248 mm)

| Attribute | Value in Document | Extracted value | Evaluation (CS) |
| --- | --- | --- | --- |
| Dimensions [mm] | 248 × 115 × 248 | 248 × 115 × 248 | correct (1.000) |
| Compressive strength class [–] | 12 N/mm² | 12 | correct (1.000) |
| Thermal conductivity  [W/(mK)] | 0.99 | 0.99 | correct (1.000) |
| Bulk density class [–] | 1.8 | 1.8 | correct (1.000) |
| Fire reaction class [–] | A1 | A1 | correct (1.000) |
| Weighted sound insulation index Rw [dB] | 49.9 dB *(wall thickness 115 mm)* | 49.9 | correct (1.000) |
| Characteristic masonry compressive strengthfk [N/mm²] | 7.0 | 7.0 | correct (1.000) |
| Hazardous substances (contents) | Quartz  (60–95 %) | Quartz (60–95 %) | correct (1.000) |

**Product 9 (Lightweight Masonry Mortar LM21**): In this case, durability was not expressed as a class (frost resistance F1/F2) but rather as a textual description (“suitable for moderately aggressive environments”). The initial prompt expected a class and therefore correctly returned None. After adapting the prompt to accommodate the textual form, the correct statement was extracted. Such prompt adjustments were also made for other fields whenever documentation deviated from the normative presentation—for example, when no mortar class was specified but a quality description was provided instead. Ultimately, for Product 9—as for all other test products—all intended attributes were correctly extracted; missing entries were left blank. The system also reliably identified specialized information such as chromium content (≤ 0.1%), VOC emissions (0.0 g/L), or indirectly stated workability time. The self-consistency majority vote further ensured that outlier responses were not adopted as final results.

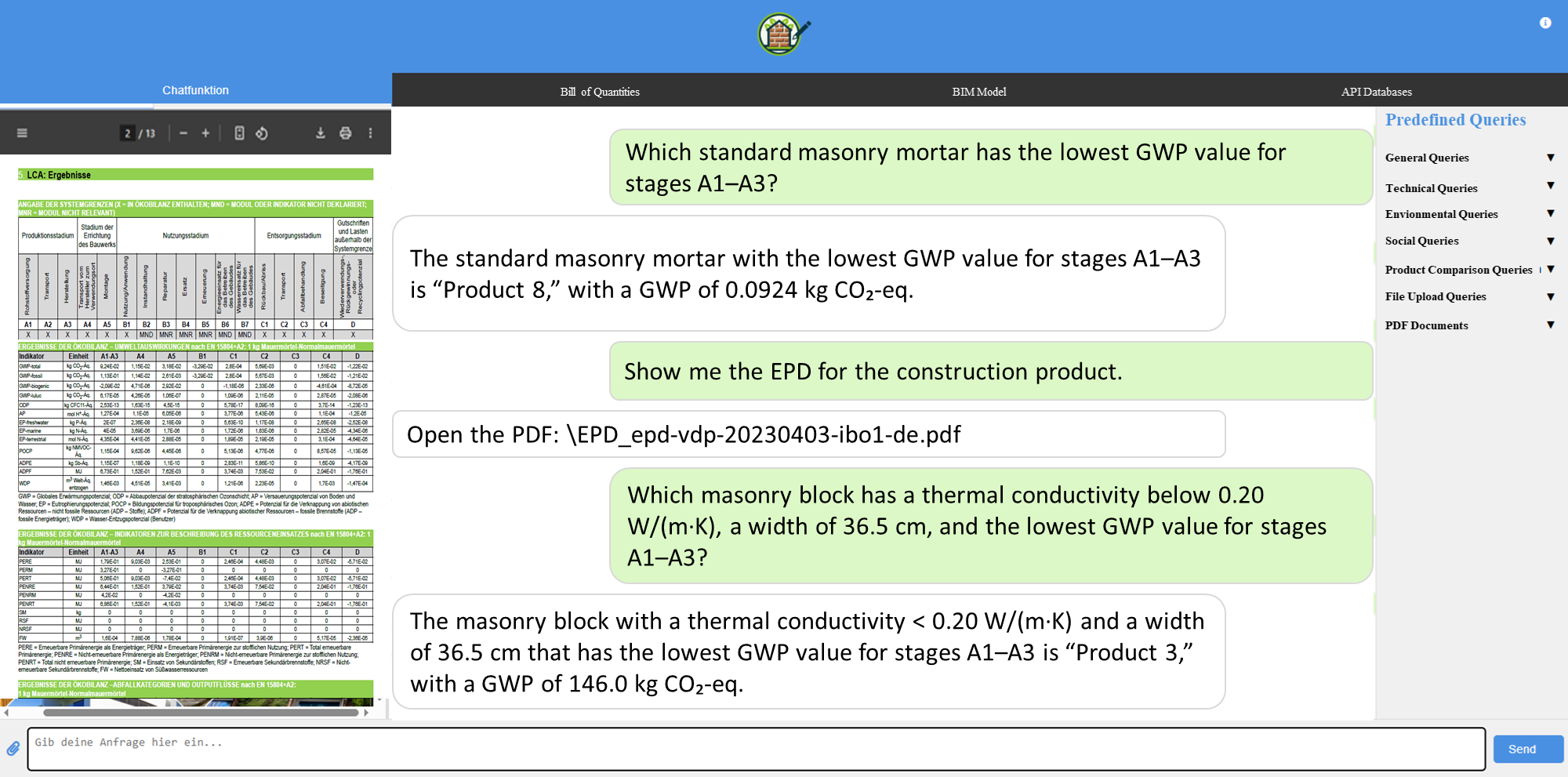
For each product, the final assigned product type, the number of extracted fields by category (technical / ecological / socio-cultural; economic not applicable), the extraction accuracy, and the average confidence score are reported. Of a total of 1,431 extracted fields (156 technical and 1,275 ecological), 99.1% were correctly extracted. In absolute terms, only two incorrect values occurred—one initially misinterpreted entry before prompt optimization, and one rounding error, which was corrected through self-consistency voting. On average, approximately nine technical and 124 ecological data points were captured per product—the latter comprising detailed EPD indicators for each life-cycle module. Socio-cultural information was sparse, averaging fewer than one field per product (primarily hazardous substance declarations from SDSs). The average confidence score was 0.98. In eight out of ten cases, the system achieved full agreement (score 1.0); in two mortar products, the score was 0.9, indicating isolated uncertainties (consistent with the prompt adjustments described above). Automatic product type classification matched the actual type in all cases (see Section 3.1). These results underscore the performance of the GraphRAG approach: despite heterogeneous document structures and varying content, virtually all required information could be accurately identified, normalized, and structured, provided it was contained in the manufacturer-supplied documentation. This ensures a robust data foundation for subsequent analyses.

## Runtime and Cost Profile of Extraction

To assess operational efficiency, the extraction pipeline was evaluated on the examined sample (10 products, 50 variants) with respect to runtime, token consumption, and costs. Across a total of 7,283 API requests, the wall-clock time was 1 h 59 min. The total token usage amounted to 11,837,326 input tokens and 193,786 output tokens (≈ 61:1), resulting in total costs of USD 1.21 (≈ EUR 1.04). On average, this corresponds to 11.9 minutes per product and 2.38 minutes per variant, with costs of USD 0.121 per product (USD 0.024 per variant). Per request, approximately 1,652 tokens were processed on average. These metrics demonstrate that the pipeline combines the high content accuracy reported in Section 3 with low resource consumption, making it well-suited for scalable deployment in planning and verification workflows.

# Queries in Decision Support Systems

Following the extraction of product variants and attributes, the results are compared against the sustainability criteria of various assessment systems (Step 5 in Fig. 1) and stored in a knowledge graph database (Step 6). For the research project, Neo4j was selected as the knowledge Graph platform. The stored product information can be queried using Cypher, Neo4j’s query language. To enable user-friendly interaction, a web-based decision support system was developed. This tool allows designers, contractors, and sustainability auditors to pose natural-language questions, which are translated via the OpenAI API into Cypher queries.



**Figure 2** Web interface of the GraphRAG pipeline as a decision support system

The graph is queried in real time, and the results are thentranslated back into comprehensible natural-language responses. Figure 2 shows the web interface of the GraphRAG pipeline. In the example, the system answers two questions:

* Question 1: “Which general-purpose masonry mortar has the lowest GWP value for phase A1–A3?” (single criterion query)

Answer 1: “The general-purpose masonry mortar with the lowest GWP value for phase A1–A3 is Product 8 with a GWP of 0.0924 kg CO₂-eq.”

* Question 2: “Which masonry unit has a thermal conductivity of < 0.20 W/(mK), a width of 36.5 cm, and the lowest GWP value for phase A1–A3?” (Multi-criteria)

Answer 2: “The masonry unit with a thermal conductivity lower than 0.20 W/(mK), a width of 36.5 cm, and the lowest GWP value for phase A1–A3 is Product 3 with a GWP value of 146 kg CO₂-eq.”

Following the first question, the command “Show me the EPD of the construction product” displayed the corresponding EPD in the integrated PDF viewer, allowing the reported data to be directly verified against the primary source. This example illustrates that the tool operates on reliable data and effectively supports designers, contractors, and sustainability auditors in selecting and documenting sustainable construction products through the automated extraction and provision of information.

# Conclusion

This study presents a GraphRAG pipeline that automatically transforms unstructured construction product documents—technical data sheets, safety data sheets, environmental product declarations, and declarations of performance—into a queryable knowledge base. After standards-guided preprocessing, PDFs are converted into structured text and tables, and EPDs are mapped to a consistent JSON schema.

A two-stage classification process, combining semantic embeddings with LLM-based chain-of-thought validation, assigns each product precisely to its type (100% accuracy). A second LLM identifies all product variants and extracts type-specific attributes, while a self-consistency ensemble merges query results and harmonizes units. Applied to ten masonry products, the pipeline achieved over 99% correct extraction of technical, ecological, and socio-cultural attributes—without hallucinations.

The extracted data, including requirements from major assessment systems (EU Taxonomy, DGNB, QNG), are stored in a Neo4j knowledge graph for complex Cypher queries. A web interface translates natural language into graph queries, enabling designers, contractors, and sustainability auditors to filter and compare products by criteria such as compressive strength, thermal conductivity, or global warming potential.

The results demonstrate that retrieval-augmented generation, combined with knowledge graphs, can unlock previously inaccessible document data with high precision, providing a robust, time-efficient foundation for data-driven, sustainable decision-making in masonry construction.

References

[1] United Nations Environment Programme (2024). Global Status Report for Buildings and Construction: Beyond foundations: Mainstreaming sustainable solutions to cut emissions from the buildings sector. Nairobi. https://doi.org/10.59117/20.500.11822/45095.

[2] Alam Bhuiyan, M. M., and Hammad, A., A Hybrid Multi-Criteria Decision Support System for Selecting the Most Sustainable Structural Material for a Multistory Building Construction, Sustainability, MDPI, 15(4), 3128, February 2023.

[3] Theilig, K., Lourenço, B., Reitberger, R. et al. Life cycle assessment and multi-criteria decision-making for sustainable building parts: criteria, methods, and application. Int J Life Cycle Assess 29, 1965–1991 (2024). https://doi.org/10.1007/s11367-024-02331-9

[4] T. Xiao and P. Xu, “Exploring automated energy optimization with unstructured building data: A multi-agent based framework leveraging large language models,” Energy and Buildings, vol. 322, p. 114691, 2024, doi: 10.1016/j.enbuild.2024.114691.

[5] Snyder, J.; Menard, A.; Spare, N. (2018) Big Data = Big Questions for the E&C Industry. PlanGrid & FMI Corp. Research Report, April 2018.

[6] Lewis, P.; et al. (2020) Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Proceedings of NeurIPS 2020, pp. 9459 – 9474.

[7] H. Han, Y. Wang, H. Shomer, K. Guo, J. Ding, Y. Lei, M. Halappanavar, R. A. Rossi, S. Mukherjee, X. Tang, Q. He, Z. Hua, B. Long, T. Zhao, N. Shah, A. Javari, Y. Xia, and J. Tang, “Retrieval-Augmented Generation with Graphs (GraphRAG),” arXiv:2501.00309, 2025, doi: 10.48550/arXiv.2501.00309.

[8] A. Saka, R. Taiwo, N. Saka, B. A. Salami, S. Ajayi, K. Akande, and H. Kazemi, “GPT models in construction industry: Opportunities, limitations, and a use case validation,” Developments in the Built Environment, vol. 17, Art. no. 100300, 2024, doi: 10.1016/j.dibe.2023.100300.

[9] "STLB-Bau Online," DIN-Bundesanzeiger Verlag GmbH. [Online]. Available: https://www.stlb-bau-online.de/. [Accessed: Aug. 10, 2025].

[10] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, “Self-Consistency Improves Chain of Thought Reasoning in Language Models,” in Proc. 11th Int. Conf. Learn. Representations (ICLR), Kigali, Rwanda, May 1–5, 2023. [Online]. Available: https://openreview.net/forum?id=1PL1NIMMrw [Accessed: Aug. 08, 2025].

[11] Dietterich, T.G. (2000) Ensemble Methods in Machine Learning. In: Proceedings of MCS 2000, Lecture Notes in Computer Science 1857, pp. 1 – 1