

CHAPTER 2 - LITERATURE REVIEW

2.1 Introduction

PT XYZ is a leading player in Indonesia's heavy equipment industry, primarily involved in the production, importation and distribution of heavy machinery and spare parts in Indonesia. With a diverse range of products and spare parts including up to 200,000 distinct part numbers (PN) actively imported, the company plays a vital role in supporting sectors such as mining, construction, and agriculture. Spare parts not only ensure the smooth operation of heavy machinery but also minimize downtime, extend the lifespan of equipment, and enhance overall productivity. Figure 2.1 show how the heavy machinery operation almost nonstop at a jobsite. Given the high cost and strategic value of heavy machinery, any delay in the supply chain can result in significant operational disruptions and financial losses.



Figure 2.1 An excavator loads soil into haul truck at a mining site, showcasing heavy equipment in action.

Accurate HS code classification is essential for ensuring that these parts are imported correctly, incurring the appropriate tariffs, and avoiding unnecessary delays that could impact the maintenance and performance of heavy equipment. The classification of

goods under the Harmonized System (HS) is a key part of global trade. It helps companies follow customs rules, reduces shipping delays, and lowers the risk of fines due to wrong classifications. In industries like heavy equipment, where product descriptions are often complex and detailed, accurate classification is critical for smooth operations and meeting government rules.

In Indonesia, the process is guided by the *Buku Tarif Kepabeanan Indonesia* (BTKI), which integrates the global HS framework with the regional ASEAN Harmonized Tariff Nomenclature (AHTN). BTKI also includes special rules under the *Ketentuan Umum untuk Menginterpretasi Harmonized System* (KUMHS). As shown in Figure 2.1, the hierarchical structure of BTKI provides a standardized way to classify goods accurately, helping companies stay consistent with national and regional trade policies.

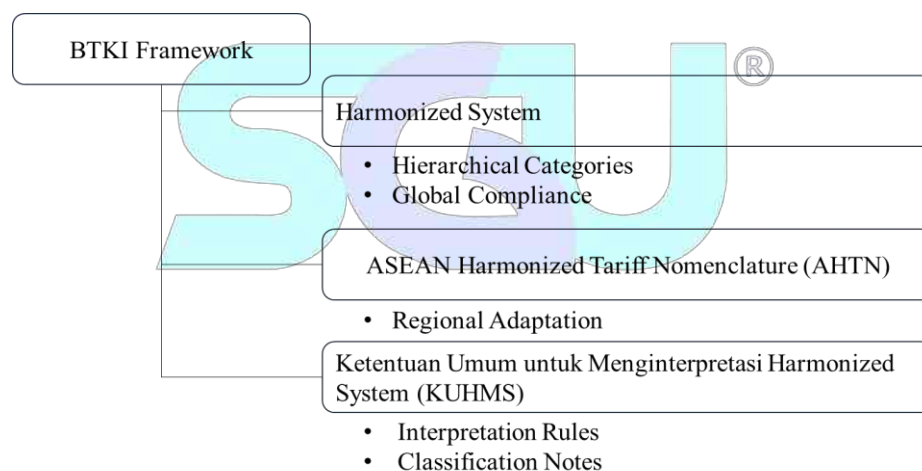


Figure 2.2 The Hierarchical BTKI Framework

Traditional classification methods often depend on manual checks and expert judgment. These methods can be time-consuming, inconsistent, and difficult to scale. The complexity of heavy equipment part descriptions makes the problem worse, increasing the likelihood of wrong classifications, which can disrupt supply chain operations and lead to extra costs. Emerging technologies such as machine learning (ML), knowledge graphs, and large language models (LLMs) offer transformative solutions. They boost

classification accuracy, provide transparent reasoning, and improve efficiency, aligning well with the structured and contextual approach required by the BTKI framework.

This chapter critically examines existing literature on HS Code classification. It covers traditional methods, explores the latest developments in ML, and discusses how knowledge graphs and LLMs can be integrated. By identifying the weaknesses of current approaches and highlighting gaps in the BTKI framework, this chapter lays the groundwork for a new system that applies advanced technologies to address classification challenges.

2.2 Problem Overview

The Harmonized System (HS) classification is a critical process for global trade, ensuring accurate identification of goods for customs duties, trade statistics, and adherence to regulations. Despite its importance, the HS system poses several challenges due to its inherent complexity, dependence on traditional methods, and the constant evolution of international trade.

2.2.1 Import Process and HS Code Classification

The import process begins with an order from a customer or supplier. The exporter ensures the goods are appropriately packed, sealed, and accompanied by necessary documentation, such as the commercial invoice, packing list, Bill of Lading (B/L) or Air Waybill (AWB), certificate of origin, and insurance. These documents are critical for assigning the correct Harmonized System (HS) code, which determines the applicable tariffs and duties. Upon arrival at the destination port, customs officials verify the importer's details, goods descriptions, and HS codes. Inspections ensure compliance with regulations and confirm duty assessments. Once duties are paid, shipments are cleared for delivery to their final destination, such as a warehouse or bonded facility. The process, illustrated in Figure 2.3, highlights the importance of accurate HS code assignment. Misclassification can lead to financial penalties, delays, and additional administrative burdens.

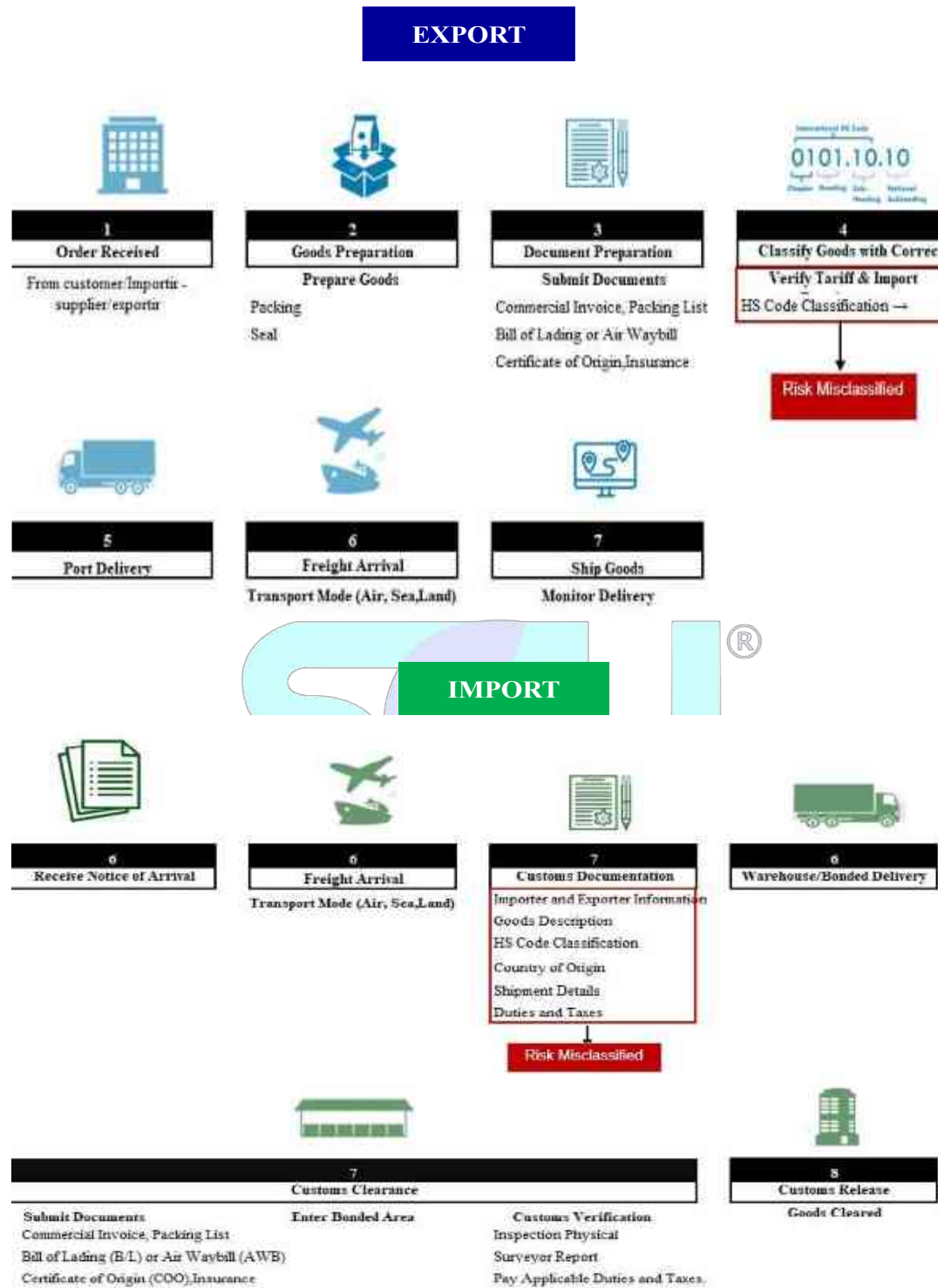


Figure 2.3 Flow process illustrating key steps in HS code classification during the import-export process in Indonesia.

The company frequently encounters challenges in accurately classifying imported goods under the appropriate HS codes, leading to potential financial and operational consequences. Incorrect HS code classification can lead to severe financial and non-financial impacts for companies engaged in international trade.

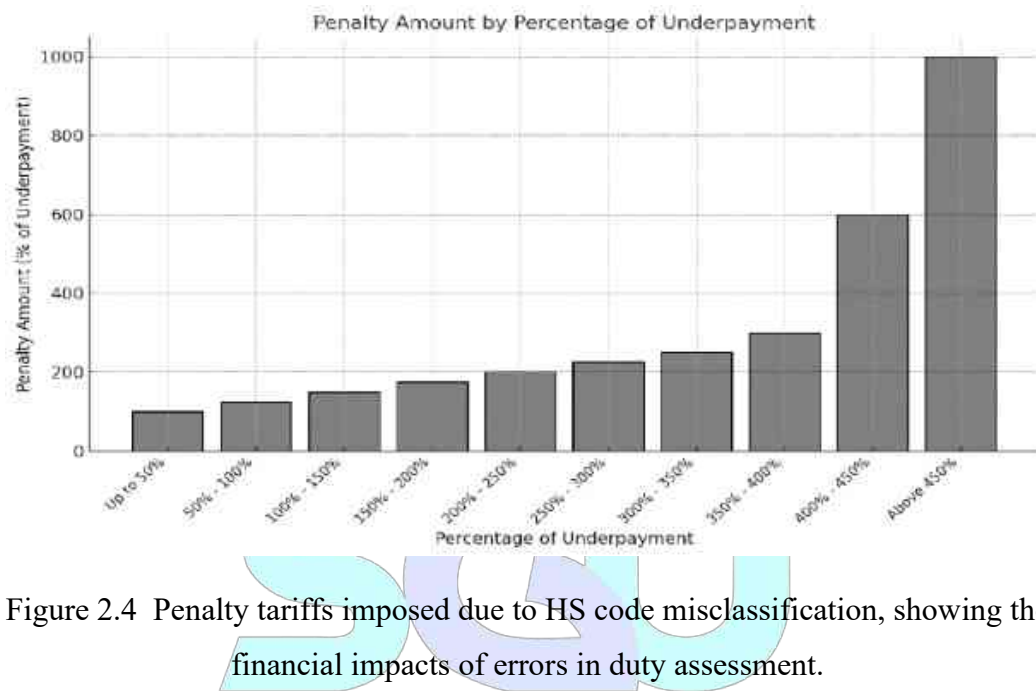


Figure 2.4 Penalty tariffs imposed due to HS code misclassification, showing the financial impacts of errors in duty assessment.

Financially, misclassification can result in significant penalties, with underpayment fines escalating to 100%-1000% of the unpaid amount based on the severity of the error as shown in Figure 2.4, in cases like PT XYZ, duties alone can incur fines of up to \$1.0 million USD, \$0.11 million USD for general taxes and \$0.09 million USD for withholding taxes (see Figure 2.5). Additionally, in customs clearance, caused by incorrect HS codes triggering manual verification or physical inspections, leading to demurrage and storage fees of ~\$100-\$200/day per container, as well as penalties for missed deadlines, which could be around 5%-10% of the contract value. Overpayment from tariff misclassifications, often resulting in duties 5% higher than necessary, further erodes profitability.



Figure 2.5 Financial impacts of HS code misclassification on PT XYZ, including penalties, overpayment, and increased costs from delays.

Non-financial impacts also carry substantial consequences:

1. Misclassification can lead to non-compliance with import regulations, causing rejection of customs declarations, seizure of goods and potential suspension of import permits.
2. Repeated HS code errors can flag a company as high-risk by customs authorities, resulting in increased stricter inspections and delays for shipments. Furthermore, the company's reputation with customs and stakeholders may negatively impacting future operations.
3. Disruption in the supply chain due to delays not only affects production schedules but also impacts to customer satisfaction, potentially leading to lost business. Legal risks also pose a significant non-financial burden, with incorrect HS code submissions possibly leading to allegations of customs fraud, resulting in costly and lengthy legal disputes.
4. Finally, frequent audits due to repeated HS code errors impose a considerable operational burden, slowing down clearance processes and increasing administrative overhead.

Both financial and non-financial impacts emphasize the importance of accurate HS code classification in maintaining smooth and compliant international trade operations.

2.2.2 Regulatory Changes Impact on Heavy Equipment Supply Chains

One of the most significant challenges in Harmonized System (HS) code classification is keeping pace with evolving trade regulations. Frequent updates can disrupt supply chains, increase compliance burdens, and pose financial risks for businesses that classify their goods incorrectly. Indonesia's *Buku Tarif Kepabeanan Indonesia* (BTKI) is periodically revised to reflect global HS amendments and respond to emerging sectors. BTKI 2022, the latest update, expanded tariff lines from 10,841 (BTKI 2017) to 11,552, creating new classifications for advanced machinery and renewable energy components. Table 2.1 summarizes the key milestones in BTKI's evolution, illustrating how each iteration aligns with international trade standards and responds to shifts in technology and industry.

Table 2.1 Historical Updates of BTKI

Version	Year	Key Updates	Impact
BTKI 2007	2007	First major adoption of the ASEAN Harmonized Tariff Nomenclature (AHTN) integrated with HS 2007.	Improved regional trade consistency under ASEAN agreements.
BTKI 2012	2012	Updated to align with HS 2012 revisions.	Enhanced classifications for emerging industries, e.g., electronics and environmental goods.
BTKI 2017	2017	Aligned with HS 2017 and AHTN 2017.	- Expanded tariff lines to 10,841. - Improved classifications for high-tech goods and renewable energy parts.
BTKI 2022	2022	Aligned with HS 2022 and AHTN 2022. Tariff lines increased to 11,552. New chapters added for emerging sectors (e.g., renewable energy and advanced technologies).	- Modernized trade facilitation with detailed classifications. - Supported economic priorities like electric vehicles and green energy.

Beyond BTKI updates, Indonesia has enacted increasingly strict regulations for importing certain categories of goods, notably iron, steel, and their derivatives. Building on earlier legal frameworks—such as Regulation No. 54/M-DAG/PER/12/2010 and Regulation No. 113/M-DAG/PER/12/2015 Minister of Trade Regulation No. 25 of 2022 introduces additional licensing requirements, more stringent product

documentation, and tighter controls on Harmonized System (HS) code classification for iron and steel commodities. These measures, have significant implications for companies like PT XYZ that rely heavily on importing steel components for their heavy equipment operations. Figure 2.5 illustrates examples of parts classified as restricted iron and steel derivatives. Although the government's policy assumes that fasteners such as bolts, nuts, and washers can be sourced from local industries, it often overlooks the specialized materials and high-quality standards demanded by heavy equipment applications. As a result, relying solely on domestic suppliers may be neither feasible nor sufficient for meeting the stringent requirements of these components.



Figure 2.6 Iron and steel hardware parts including bolts, nuts, and washers commonly used in heavy equipment assembly and maintenance.

PT XYZ imports a wide range of heavy equipment parts, many of which contain iron or steel. Under Regulation No. 25 of 2022, importers must now navigate a more detailed and stricter permitting process that requires precise and accurate HS code classification for each product. This is particularly critical because the new regulation imposes increased scrutiny on import documentation, aiming to prevent illicit or improper entry of steel-related goods. The consequences of misclassification can be severe: inaccuracies in HS code reporting may lead to delays at customs, penalty fees, or in

worst-case scenarios, the revocation of import licenses. For PT XYZ, any disruption or additional cost in obtaining the necessary steel components can have cascading effects on production schedules and overall operational efficiency.

Furthermore, the heightened documentation obligations under the new regulation demand that PT XYZ maintain thorough records to demonstrate compliance with product standards and import criteria. This entails greater administrative effort, from compiling product certifications to verifying the technical specifications of each imported part. Ensuring robust internal controls and training staff on correct HS code classification becomes essential to mitigate risks. Failure to meet the new licensing and documentation standards not only carries reputational damage but also introduces the possibility of stricter government audits and potential legal repercussions.

In summary, while Regulation No. 25 of 2022 is designed to protect Indonesia's domestic steel industry and ensure product integrity, it imposes additional layers of complexity on import-dependent companies like PT XYZ. The regulation underscores the critical need for accurate HS code classification, as errors can trigger severe regulatory and financial consequences. For PT XYZ, adapting internal processes and compliance strategies to align with the new requirements is imperative to maintain a smooth import flow, uphold production timelines, and sustain competitiveness in Indonesia's heavy equipment market.

2.2.3 Hierarchical Complexity

The HS Code, which underpins customs tariff classification worldwide, also forms the basis for Indonesia's BTKI. Its tiered structure 21 sections, 96 chapters, and thousands of headings and subheadings—allows for a detailed breakdown of goods but increases the difficulty of achieving precise classifications. BTKI, as an adaptation of the HS Code, adds further layers of detail, making the classification process more demanding.

The hierarchical design follows a tiered approach, as shown in Figure 2.7:

1. Sections: Broad product groups (e.g., Section XVI: Machinery and Mechanical Appliances).

2. Chapters: Narrower categories within sections (e.g., Chapter 84: Nuclear Reactors, Boilers, Machinery, and Mechanical Appliances).
3. Headings and Subheadings: Specific groups of goods (e.g., Heading 8412: Other Engines and Motors; Subheading 8412.31: Hydraulic Power Engines and Motors, Linear Acting (Cylinders)).

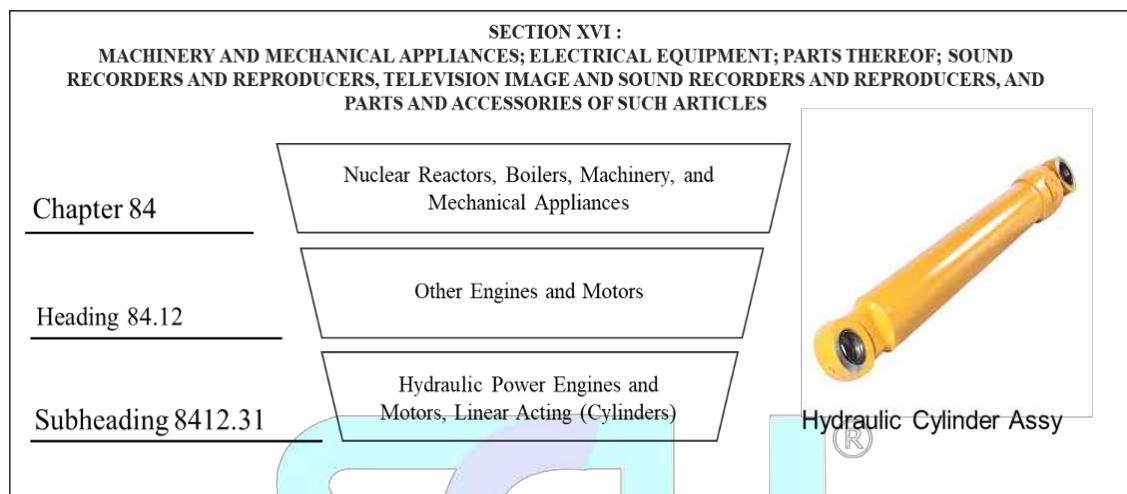
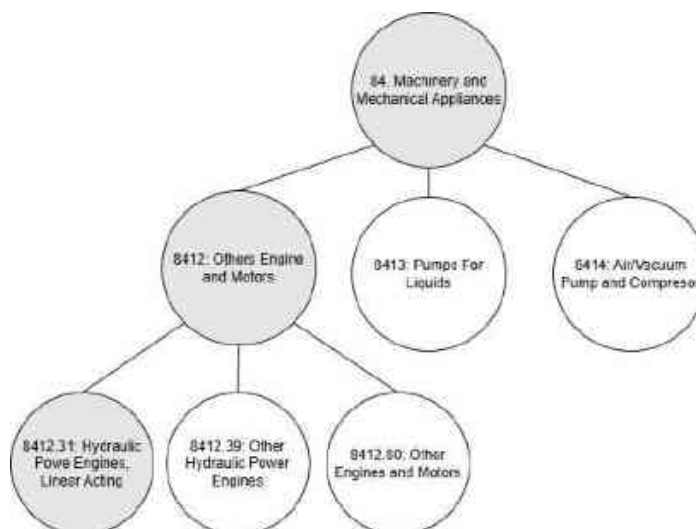


Figure 2.7 Hierarchical HS Code for PN Cylinder Assy

The diagram in Figure 2.8 illustrates the hierarchical structure of the HS Code classification system, emphasizing the relationships between various categories under Section 84, "Machinery and Mechanical Appliances."

1. Section Level (84)
At the highest level, Section 84 covers machinery and mechanical appliances. This broad category provides a starting point for classifying different types of mechanical goods.
2. Heading Level (8412, 8413, 8414):
Within Section 84, headings such as 8412 ("Other Engines and Motors"), 8413 ("Pumps for Liquids"), and 8414 ("Air/Vacuum Pumps and Compressors") further refine classification based on machinery type or application.
3. Subheading Level: The heading 8412, "Others Engines and Motors," is further divided into more specific subheadings:
 - 8412.31: Hydraulic power engines, linear acting.

- 8412.39: Other hydraulic power engines.
- 8412.80: Other engines and motors.



Pos Tarif HS Code	Description of Goods
8411.21.00	-- Of a power not exceeding 1,100 kW
8411.22.00	-- Of a power exceeding 1,100 kW
8411.81.00	-- Of a power not exceeding 5,000 kW
8411.82.00	-- Of a power exceeding 5,000 kW
8411.91.00	-- Of turbo-jets or turbo-propellers
8411.99.00	-- Other
84.12	Other engines and motors.
8412.10.00	- Reaction engines other than turbo-jets
8412.21.00	-- Linear acting (cylinders)
8412.29.00	-- Other
8412.31.00	-- Linear acting (cylinders)
8412.39.00	-- Other
8412.80.00	- Other
8412.90	- Parts :
8412.90.10	-- Of engines of subheading 8412.10
8412.90.20	-- For steam or other vapour power units incorporating boilers
8412.90.90	-- Other

Figure 2.8 Hierarchical structure of HS Code for chapter 84 Machinery and Mechanical Appliance in BTKI Regulation

2.2.4 Part Numbers and Their Structure

In the heavy equipment industry, Part Numbers (PNs) are deliberately structured to encode critical information about each component—ranging from equipment type and functionality to material composition and even manufacturing details. Unlike in many other sectors, where PNs might be assigned randomly or simply generated in sequence,

this rigorous method provides greater precision in identifying and classifying parts, as illustrated in Figure 2.9 for PT XYZ's heavy equipment.

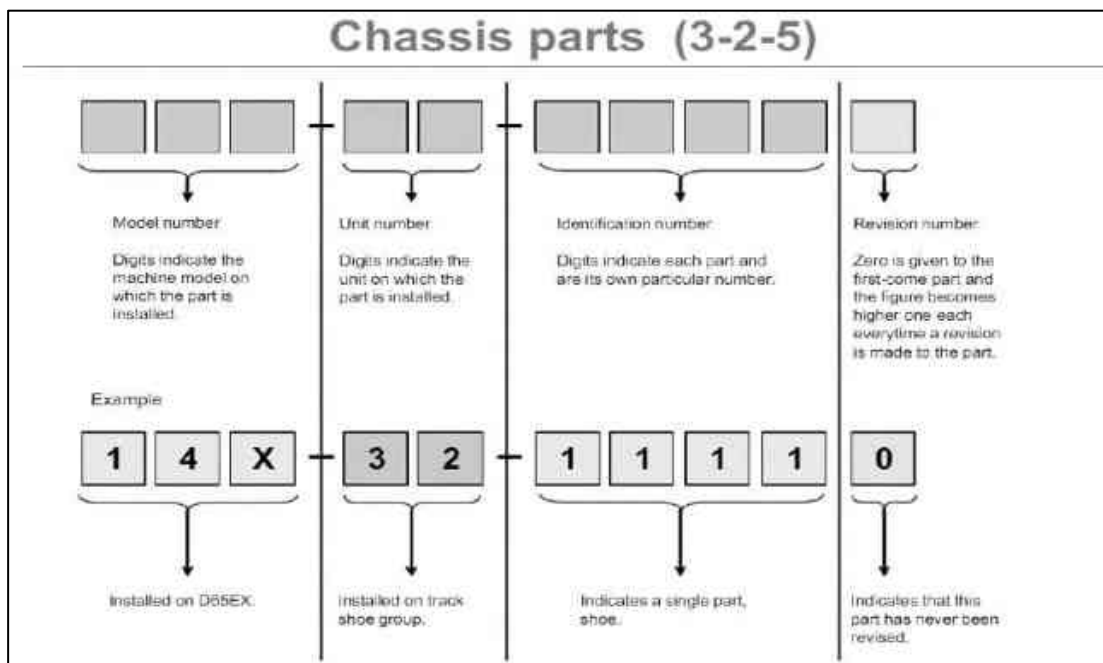


Figure 2.9 Structure PN Heavy Equipment PT XYZ systematically organized essential detailed.

This structured framework not only simplifies interpretation and categorization of parts but also offers significant benefits in machine learning applications. By extracting key features embedded within these PNs such as the equipment type, unit number, or material specifications machine learning models can more accurately predict the correct HS code. This feature extraction minimizes the risk of misclassifications that can otherwise trigger operational disruptions or increased costs due to inaccurate documentation and shipping. As a result, the systematic organization of PNs enhances both day-to-day part management and the reliability of automated HS code classification, ultimately improving overall operational efficiency for PT XYZ.

2.3 Traditional Methods for HS Code Classification

2.3.1 Reliance on Tacit Knowledge

Historically, customs officers and traders have depended on tacit knowledge acquired informally through on-the-job experience rather than structured, rule-based training

(Edirisinghe, 2012). Table 2.2 illustrates real-world misclassifications at PT XYZ, each resulting in added costs, customs penalties, or project disruptions.

Table 2.2 XYZ Misclassification HS Code

Part Name	Incorrect HS Code	Incorrect Classification Description	Correct HS Code	Correct Classification Description	Reason for Misclassification	Consequences
Air Filters for Hydraulic Excavators	8421.39.90	Other machinery and mechanical appliances; parts thereof	8421.31.00	Filtering or purifying machinery and apparatus for liquids	Misinterpreted as a general air filter without specifying application in hydraulic machinery.	Higher duty rate (10% instead of 5%), resulting in additional costs.
Fuel Injectors for Diesel Engines	8412.90.90	Other parts of engines and motors	8409.99.90	Parts for compression-ignition internal combustion piston engines (diesel or semi-diesel)	Misclassified under a generic engine part code, not specific to fuel injectors.	Increased inspection and scrutiny by customs, causing delays and additional administrative costs.
Engine Oil Filters	8421.99.00	Parts of other machinery and apparatus for filtering or purifying liquids or gases	8421.23.00	Oil or petrol-filters for internal combustion engines	Misclassified under generic machinery parts instead of specific engine oil filters.	Customs penalties due to incorrect duty assessment, increasing costs by 8%.
Hydraulic Pump for Bulldozers	8413.50.9	Pumps for liquids, other centrifugal pumps	8413.60.00	Other rotary positive displacement pumps	Lack of specificity regarding hydraulic pump use in heavy equipment.	Delay in customs clearance (two-week hold) causing supply chain disruption.

This reliance often introduces subjective interpretations and can yield inconsistent outcomes. For instance, a steel-reinforced rubber hose might be classified under hydraulic system components by one officer or by its rubber properties by another, leading to confusion and delays. Such discrepancies restrict from insufficient engagement with the rules in KUMHS, making the classification process both time-consuming and error prone. Relying solely on human expertise can create

inconsistency, slowdowns, and misunderstandings between companies and customs officials. Misclassifications disrupt timelines, inflate costs due to improper duties, and may expose businesses to legal challenges. Consequently, there is a growing push for systematic, rule-based approaches and AI-driven solutions that unify the interpretation of HS classification rules, minimize errors, and ensure smoother trade operations (Kongnakorn et al., 2023).

2.3.2 KUHMS

Traditional HS Code classification typically relies on manual processes and human expertise, often referred to as tacit knowledge. *Ketentuan Umum untuk Menginterpretasi Harmonized System* (KUMHS) provides six general rules to guide this process, focusing on initial determinations, defining goods' essential characteristics, resolving ambiguities, and applying additional guidelines for specific scenarios. These rules aim to ensure accurate and consistent classification, especially for complex products. Figure 2.10 shows the standard procedural steps typically followed when classifying goods under KUMHS, illustrating the legal principles that help determine the most suitable code (Edirisinghe, 2012). The Harmonized System (HS) classification relies on a structured and standardized approach outlined in six general rules. These rules ensure consistency and accuracy in classifying goods for customs and trade purposes. Each rule provides specific guidance, addressing different scenarios that may arise during the classification process. A summary of these rules is presented in Table 2.2.

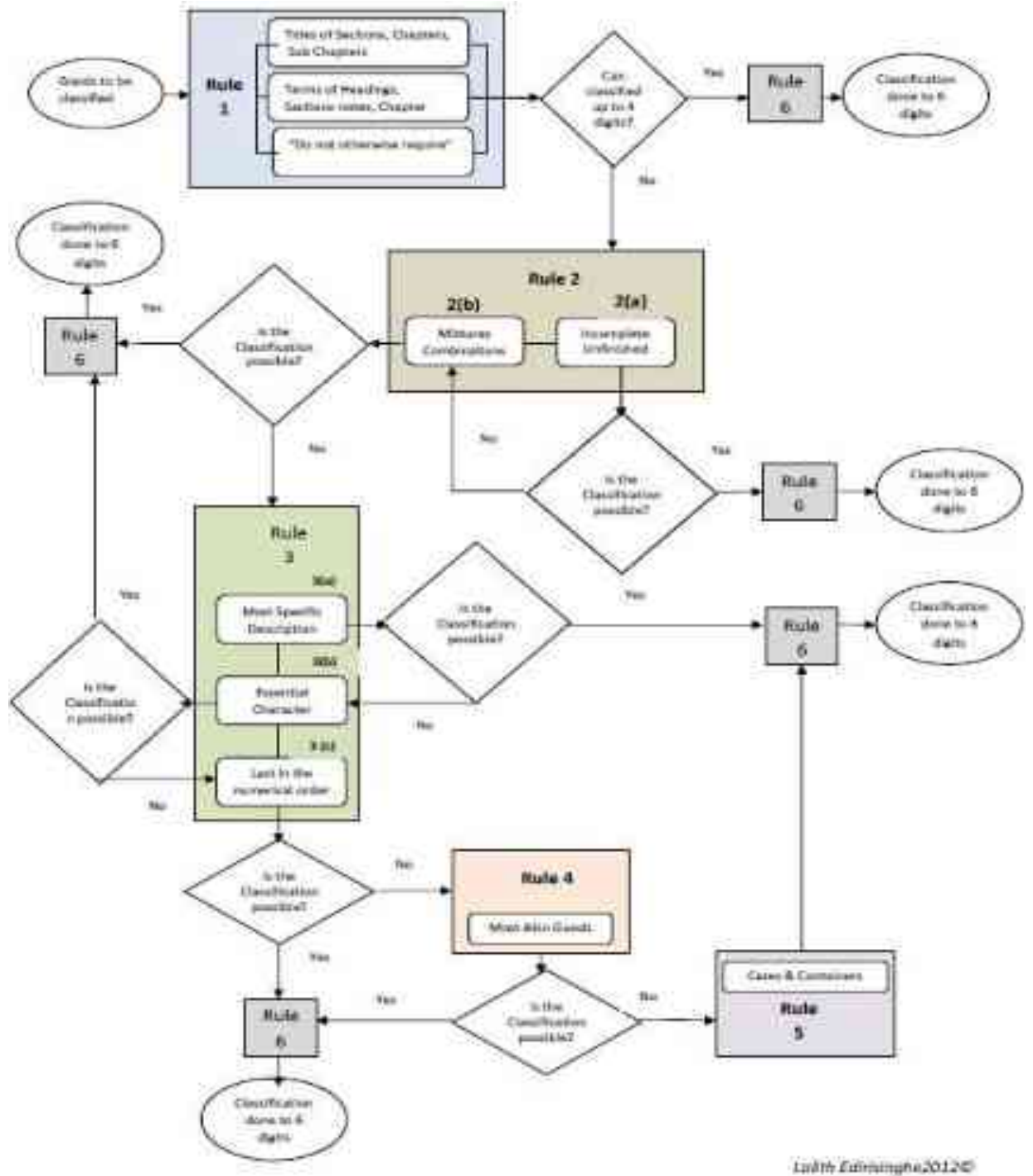


Figure 2.10 The Process Flow Chart for KUHMS Classification Goods

Table 2.3 Six General Rules for Classification Harmonized System (KUHMS)

Group	Rule	Description	Example
Initial Determination	Rule 1	Classify goods based on headings and legal notes in the HS Code.	A plastic fork is classified under 3924 (Tableware and kitchenware of plastics) because the heading clearly describes it.
	Rule 2 (a)	Incomplete or unassembled goods are classified as if they are complete if they have the essential character of the final product.	A disassembled bicycle is classified as a complete bicycle.
Goods Characteristics	Rule 2 (b)	Mixtures or combinations are classified based on the material/component that gives the essential character.	A steel spoon with a wooden handle is classified as 8215 (Spoons and forks) because steel defines its main function.
	Rule 3 (a)	When multiple headings apply, choose the most specific one.	A leather wallet is classified under 4202 (Articles of leather) rather than a general heading for other leather goods.
Resolving Ambiguities	Rule 3 (b)	For mixed materials, classify based on the component giving the essential character.	A watch with a leather strap is classified under the heading for the watch.
	Rule 3 (c)	If classification is unclear, use the heading that appears last in numerical order.	A 50/50 rye-barley mixture is classified under the later heading of the two options.
Additional Rules	Rule 4	If no heading applies, classify goods under the closest similar item.	A hybrid material is classified under the closest comparable product.
	Rule 5 (a)	Classify specialized containers with the goods they are designed to hold, if used long-term.	A violin case is classified with the violin.
	Rule 5 (b)	Classify packing materials with the goods, unless they have separate value or use.	A cardboard box is classified with its contents, but a durable metal container may be classified separately.
	Rule 6	Apply the same rules at the subheading level, comparing	A plastic fork under 3924 (Tableware) is further classified as 3924.10 (Tableware).

Although these principles offer a strong foundation, the manual nature of classification can be time-consuming and prone to error, particularly as the volume and variety of traded goods increase. This limitation has driven the need for more efficient methods of HS Code classification.

2.3.3 Challenge and Limitation of the Traditional HS Code Classification Method at PT XYZ

This review critically evaluates the traditional method, identifying key areas where technological support could enhance accuracy, consistency, and efficiency. The HS Code classification at PT XYZ relies on manual decision-making and tacit knowledge, ensuring regulatory compliance and operational flexibility under KUMHS guidelines. However, as trade volumes grow and regulations tighten, this approach reveals inefficiencies, inconsistencies, and scalability challenges. While human expertise remains valuable, increasing trade complexity demands a more structured and efficient classification process, highlighting the need for technological support to enhance accuracy, consistency, and efficiency.

Institutional Knowledge and the Challenge of Consistency

Traditional manual classification benefits from deep institutional knowledge, allowing customs officers to interpret complex product descriptions and ambiguous classifications (Edirisinghe, 2017). However, reliance on tacit knowledge leads to inconsistencies, where similar products may receive different classifications based on individual judgment. PT XYZ has faced misclassifications, incorrect duty assessments, and increased customs scrutiny, highlighting the need for standardized decision-making frameworks to ensure regulatory compliance and consistency.

Flexibility vs. the Need for a Systematic Approach

The flexibility of human interpretation allows customs officers to factor in product specifications, applications, and industry-specific nuances that rigid rule-based systems might overlook. This adaptability is especially beneficial for specialized heavy equipment parts, where classification may not be straightforward. However, without a clear structure, this flexibility poses scalability challenges. PT XYZ's growing import

volume manually handling thousands of classifications per month not only increases the likelihood of delays and bottlenecks but also makes it difficult to track and refine past decisions. A systematic framework is needed to maintain flexibility while ensuring consistency and efficiency.

Immediate Decision-Making and the Risk of Errors

Manual classification allows for instant decisions, useful for urgent shipments and unexpected modifications. However, speed does not ensure accuracy. Fatigue, oversight, or outdated regulations increase the risk of misclassification, as seen in PT XYZ's costly errors with hydraulic excavator air filters, leading to higher duties and delays. Errors not only incur financial losses but also slow customs clearance. To maintain efficiency while reducing mistakes, decision-support tools should be considered.

Regulatory Compliance and the Burden of Documentation

Manual classification ensures compliance with BTKI regulations and supports dispute resolution. However, evolving policies like Indonesia's Regulation No. 25 of 2022 make manual documentation increasingly unsustainable. Tracking past classifications, auditing decisions, and updating records are difficult without a structured system. To meet stricter compliance demands, PT XYZ needs an organized approach to classification management.

The Growing Need for Technological Support

While human expertise is valuable, manual classification alone is no longer sustainable. Inconsistencies, inefficiencies, and compliance burdens demand structured technological support. As trade complexity increases, PT XYZ must explore solutions that enhance accuracy, streamline compliance tracking, and improve efficiency. Rather than replacing human judgment, technology should provide structure and consistency, ensuring scalability and regulatory alignment.

Table 2.4 Strength and Limitation of HS Code Traditional Method in PT. XYZ

Aspect	Strengths	Limitations	Implications
--------	-----------	-------------	--------------

FOR INDONESIA HS CODE CLASSIFICATION: A FOCUS ON HEAVY EQUIPMENT PARTS

Knowledge-Based Decision-Making	Officers apply expertise to handle ambiguous classifications.	Reliance on tacit knowledge leads to inconsistencies and misclassifications.	Risk of customs disputes, financial penalties, and delays.
Flexibility in Interpretation	Officers consider product-specific nuances and industry context.	Lack of structured guidelines results in varying classifications.	Difficulty in maintaining standardization and audits.
Scalability of the Process	Allows classification without system dependency.	Labor-intensive, slow, and inefficient with high transaction volumes.	Bottlenecks in supply chain operations and increased workload.
Speed of Decision-Making	Enables immediate decisions for urgent shipments.	No systematic verification, leading to human errors and misclassification.	Financial losses, incorrect duties, and increased compliance risk.
Regulatory Compliance	Ensures adherence to BTKI and KUMHS regulations.	Manual record-keeping is burdensome, making it hard to track past decisions.	Difficulties in compliance updates and regulatory audits.
Documentation & Traceability	Officers can justify manual classifications during audits.	No centralized database, leading to difficulty in tracking changes.	Greater risk of disputes and inefficiencies in case of audits.
Need for Technology	Human expertise remains valuable in complex cases.	Manual methods are no longer sustainable for growing trade volumes.	Requires structured support to enhance accuracy and efficiency.

In summary this table 2.4 highlights the operational inefficiencies, financial risks, and compliance challenges of PT XYZ's manual classification method, reinforcing the need for a structured technological solution to improve accuracy, scalability, and regulatory compliance. Machine learning as one of the technology solution available in classification will be discuss in next section.

2.4 Machine Learning for HS Code Classification

The challenges of manual HS Code classification including inconsistencies, inefficiencies, and growing regulatory demands underscore the need for a more

structured, scalable approach. While human expertise is valuable, reliance on manual processes is unsustainable for PT XYZ's high transaction volume.

One promising approach is machine learning (ML), which enables automated HS Code classification by analysing textual data from product descriptions. Machine learning (ML) offers a solution by automating HS Code classification through text mining, feature extraction, and classification algorithms. This section explores key ML techniques, including text mining, TF-IDF, and classification algorithms, along with their potential and limitations.

2.4.1 Text Mining Transforming Unstructured Text into Usable Data

Text mining is a foundational process in the automation of HS code classification. It involves extracting meaningful information from unstructured textual data, such as product descriptions, and transforming it into a structured format suitable for machine learning algorithms. Since product descriptions are often written in free-form text, they must undergo preprocessing and feature extraction to ensure consistency and relevance in classification tasks. Text mining provides the methods to preprocess and convert this textual data into a numerical format that machine learning algorithms can interpret (Karmakar, 2022). This preprocessing includes:

1. **Tokenization:** Breaks the text into smaller units called tokens (usually words). For example, the description "Stainless steel kitchen knife" is split into the tokens ["stainless", "steel", "kitchen", "knife"].
2. **Normalization:** Standardizes text to reduce variability caused by formatting, case differences, or superfluous characters. This includes lowercasing, punctuation removal, and stop-word removal. For instance, words like "the" and "is" are typically excluded because they add little meaning for classification.
3. **Stemming and Lemmatization:**
 - 1) **Stemming** trims words to their root forms by removing suffixes (e.g., "knives" → "knife"), which may not yield a real word.
 - 2) **Lemmatization** maps words to their base forms using vocabulary and grammar rules (e.g., "knives" → "knife"). Because lemmatization preserves word meaning more precisely, it is often preferred for English-language tasks.

Machine learning models rely on clean, structured data to identify relevant patterns rather than getting bogged down by noise—such as extra punctuation, uppercase variations, or filler words. By applying text mining techniques, we ensure that product descriptions are transformed into a consistent, relevant format that highlights the core attributes (e.g., “stainless steel”). As a result, models can more accurately interpret the information and improve the likelihood of classifying products into the correct HS codes (Karmakar, 2022).

2.4.2 TF-IDF: Extracting Significant Features

Machine learning models fundamentally operate on numerical data. By converting text into vectors, algorithms like Naive Bayes, SVM, or Random Forest can apply mathematical operations—such as calculating distances, similarities, or probabilities—to discover meaningful patterns. These vectors enable the models to learn how specific words (or combinations of words) correlate with certain HS codes, which would be impossible if the data remained in raw text form.

After preprocessing product descriptions through text mining techniques, the next step involves identifying words that carry the most significance for predicting HS codes. Term Frequency-Inverse Document Frequency (TF-IDF) is a widely used technique that quantifies the importance of words within a document relative to a collection of documents (corpus). By transforming unstructured text into numerical representations, TF-IDF enables machine learning algorithms to process and classify product descriptions effectively. TF-IDF combines two key metrics to assign a weight to each term in a document. This weight reflects the term's relevance to that document while penalizing terms that are common across the entire dataset with following explanation:

1. Term Frequency (TF)

TF measures how often a term appears in a document. One common formula is:

$$\text{TF}(t, d) = \frac{\text{Number of times } t \text{ appears in } d}{\text{Total Numbers of words in } d}$$

For example, in “steel bolts for heavy machinery,” the term “steel” may have a relatively high TF if it appears frequently across product descriptions dealing with metal parts (Karmakar, 2022).

2. Inverse Document Frequency (IDF):

IDF measures how rare or specific a term is across all documents. A common formula is:

$$\text{IDF}(t) = \ln\left(\frac{N}{1 + \text{DF}(t)}\right)$$

Where N is the total number of documents and $\text{DF}(t)$ is the number of documents containing the term t . Terms like “Komatsu” or “hydraulic” often have higher IDF values because they appear in fewer, specialized product descriptions (Karmakar, 2022).

3. TF-IDF Score:

The TF-IDF score is calculated by multiplying a term’s TF and IDF values.

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

A high TF-IDF score indicates that a term is both common in one document and relatively rare across the entire corpus, making it an excellent feature for classification.

Why TF-IDF Was Widely Used:

1. **Compatibility:** Most traditional machine learning models, including Naive Bayes, SVM, and Random Forest, require numerical input. TF-IDF effectively converts textual data into numerical vectors.
2. **Effectiveness:** TF-IDF emphasizes unique terms in product descriptions while reducing noise from common terms, enhancing model performance.
3. **Ease of Implementation:** TF-IDF is straightforward to compute and has been a standard feature extraction method in text classification tasks.

In HS code classification for heavy equipment parts, TF-IDF generates feature vectors that highlight specialized domain terms like “steel” or “hydraulic” while downplaying common words such as “for” and “equipment.” These vectors allow machine learning models to capture the nuances of product descriptions in mathematical form, ultimately improving classification performance.

2.4.3 ML Data Collection for HS Code Classification

The collection and preparation of data are foundational steps in developing machine learning models for HS code classification. This section synthesizes insights from reviewed studies under four key aspects: data sources, scope, preprocessing, and challenges.

1. **Data Sources**, various data sources have been utilized in studies to build HS code classification models:
 - 1) **Historical Customs Data**: Studies like Muslim (2022), Paramartha, Ardiyanto and Hidayat, (2021), and Chen, et al., (2021) leveraged customs declaration records containing product descriptions, HS codes, and supplementary metadata (e.g., trade volumes and exporter/importer details). These datasets are rich in labeled data but often contain inconsistencies due to misclassifications and missing entries.
 - 2) **Product Databases**: Product catalogues from manufacturers and suppliers, as used by Paramartha, Ardiyanto and Hidayat, (2021) and Altaheri and Shaalan (2020), included domain-specific attributes such as material types, functional roles, and part numbers. These features enhanced the models' capability to capture nuanced classifications.
 - 3) **Regulatory Resources**: Official taxonomies, including the Harmonized System Nomenclature and Indonesia's BTKI framework, were integral in aligning models with hierarchical classifications. These were extensively used by Altaheri and Shaalan (2020) and (Chen, et al., 2021).
2. **Scope of Data**, the reviewed datasets differed in their depth and granularity:
 - 1) **Granularity of HS Codes**: Muslim (2022) focused on simple descriptions and high-level classifications (e.g., HS4), while Paramartha, Ardiyanto and Hidayat, (2021) addressed multiple hierarchical levels (e.g., HS2, HS4, HS6) to support scalable, multi-level models.
 - 2) **Metadata Integration**: (Chen, Wang and Jin, 2021) incorporated trade metadata (e.g., trade volumes) to contextualize HS code assignments. In contrast, Altaheri and Shaalan (2020) emphasized text descriptions, excluding metadata to focus on text-based model feasibility.

- 3) **Real-World vs. Synthetic Data:** Real-world customs data (e.g., Muslim, (2022); Chen, et al., (2021)) were valuable for practical applications but faced quality and imbalance issues. Synthetic datasets (e.g., Altaheri and Shaalan, 2020) offered cleaner, controlled inputs but lacked real-world complexity.
3. **Preprocessing, data preprocessing was critical for standardizing and cleaning raw data. Key steps included:**
 - 1) **Text Preprocessing:**
 - **Tokenization:** Commonly employed to split descriptions into tokens.
 - **Stop-word Removal:** Irrelevant words (e.g., "the", "for") were removed to focus on meaningful terms.
 - **Stemming and Lemmatization:** Used by Paramartha, Ardiyanto and Hidayat, (2021) and Altaheri and Shaalan (2020) to reduce words to their base forms (e.g., "bolts" → "bolt").
 - 2) **Feature Extraction:**
 - **TF-IDF:** Widely applied to convert text into numerical vectors, highlighting term importance.
 - **Contextual Features:** Paramartha, Ardiyanto and Hidayat, (2021) integrated bigrams (e.g., "hydraulic pump") to capture contextual relationships.
 - 3) **Data Cleaning:**
 - **Error Correction:** Chen et al., (2021) emphasized cleaning customs records, including validation by experts to rectify misclassified HS codes.
 - **Missing Data:** Handled through imputation or exclusion based on the extent of data gaps.
4. **Challenges**

Common challenges encountered across studies include:

 - 1) **Data Imbalance:** Disproportionate representation of HS codes was a prevalent issue. Paramartha, Ardiyanto and Hidayat, (2021) addressed this with oversampling and synthetic techniques such as SMOTE.
 - 2) **Errors in Historical Data:** Misclassifications in customs records hindered training. Chen, et al. (2021) mitigated this by involving domain experts for data validation.

- 3) **Preprocessing Complexity:** Tokenization, stop-word removal, and feature extraction, while essential, added considerable complexity to workflows (Muslim, 2022; Altaheri and Shaalan, 2020).

By integrating these methodologies, this section provides a comprehensive framework for effective data collection and preparation in HS code classification. Table 2.5 provides a summary of related works focused on data processing for machine learning.

Table 2.5 Related Work Data Processing for Machine Learning

Study	Method	Data and Variables	Text Processing	Key Challenges
Altaheri & Shaalan (2020)	Supervised learning (SVM, Naive Bayes)	Customs data (Dubai) - Product descriptions, HS codes, keywords	Tokenization, Stemming, TF-IDF	Struggled with hierarchical data, handling complex product descriptions, noisy data
Muslim (2022)	Supervised learning (Naive Bayes)	Indonesian Customs data (PIB) - Product descriptions, HS codes, import duties	TF-IDF, Data Cleaning	Ambiguous, incomplete descriptions, inconsistent data
Paramartha, Ardiyanto and Hidayat, (2021)	Supervised learning (Random Forest, Multinomial Naive Bayes)	Indonesian Customs data, multilingual data - Product descriptions, HS codes, language-specific features	TF-IDF, Bigrams, Data Preprocessing	Inconsistent or unclear descriptions, need for extensive preprocessing
Shubham (2023)	Ensemble-based approach (BERT, Knowledge Graphs)	Customs data, publicly available datasets - Product descriptions, HS codes, Named Entities (e.g., product names, materials)	BERT Embeddings, Named Entity Recognition (NER), Distance-based methods	Fine-grained hierarchical classification, thin-volume classes, scalability issues

Study	Method	Data and Variables	Text Processing	Key Challenges
Chen, et al.,(2021)	Unsupervised learning (Sentence embeddings)	Historical customs declaration data (Dutch customs) - Product descriptions, HS codes, historical data	Sentence embeddings, Text similarity	Lack of labeled data, adapting across different countries' HS systems
Du et al., (2021)	Hierarchical model (HScoreNet)	Historical customs data - Product descriptions, HS codes, hierarchical levels	Hierarchical classification	Real-time adaptability to dynamic HS code updates, scalability

2.4.4 Machine Learning Algorithms for HS Code Classification

The task of assigning Harmonized System (HS) codes, especially for complex and detailed product descriptions such as those in the heavy equipment industry, can benefit significantly from machine learning algorithms. Each algorithm is built on unique theoretical foundations, providing distinct strengths and trade-offs (Karmakar, 2022). This section explores the underlying concepts of these algorithms in depth, highlighting their relevance to HS code classification.

1. Naive Bayes

A probabilistic classifier rooted in Bayes' theorem, assuming conditional independence among features. With TF-IDF vectors, Naive Bayes calculates the likelihood of each token or term under different HS code categories. Its simplicity and low computational cost make it suitable for large-scale datasets. Nevertheless, its independence assumption sometimes leads to misclassifications when terms are strongly correlated.

2. Decision Tree

Decision Trees partition the feature space via successive splits, each guided by a criterion (e.g., Gini impurity or information gain). When applying TF-IDF features, these splits may be determined by whether a particular term's weight

exceeds a threshold. Decision Trees are often considered interpretable, providing transparency for regulatory contexts. However, they can overfit, especially in high-dimensional text data, if not properly pruned or constrained.

3. k-Nearest Neighbor (k-NN)

k-NN is a non-parametric method that classifies new instances by referencing the majority class among their “k” nearest neighbors in the training set. Distance metrics—often cosine similarity for TF-IDF—are used to determine the closeness of descriptions. Although k-NN can perform well for smaller, structured datasets, it tends to scale poorly because each new prediction requires distance computations against all training points.

4. Linear Support Vector Machine (SVM)

Support Vector Machines (SVMs) identify an optimal hyperplane (or set of hyperplanes) in a high-dimensional feature space, thereby maximizing the margin between distinct classes. Linear SVMs often perform robustly on text classification tasks due to TF-IDF vectors’ inherently high dimensionality (Karmakar, 2022). Non-linear kernels, such as the Radial Basis Function (RBF), allow the algorithm to capture more complex relationships among terms—a notable benefit when different HS codes share extensive vocabulary. However, successful application of SVMs frequently depends on tuning parameters like the penalty term C and kernel-specific parameters, which can be computationally intensive and data-dependent.

5. Random Forest

Random Forest is an ensemble of decision trees, each trained on bootstrapped samples of the data and, often, a random subset of features. Such randomization mitigates overfitting, resulting in enhanced robustness and accuracy. In HS code classification, Random Forest can effectively handle complex interactions among TF-IDF features, though memory usage and training times increase as the forest grows.

6. AdaBoost

AdaBoost (Adaptive Boosting) sequentially boosts weak learners—often shallow decision trees—by prioritizing instances misclassified in earlier rounds. This

iterative approach yields a strong ensemble capable of capturing intricate patterns in TF-IDF data, including class imbalances or subtle term distinctions. Nonetheless, AdaBoost can require careful tuning to avoid overfitting and can be slower to train compared to parallelizable methods such as Random Forest.

These algorithms range from simple models like Naive Bayes, which prioritize efficiency, to complex ensemble techniques like Random Forest and AdaBoost, which aim for accuracy and robustness. Simpler algorithms are often sufficient for broad HS code categorization tasks, while more advanced methods are suited to nuanced classifications requiring precision and the ability to handle ambiguous or noisy data. By aligning algorithm choice with task complexity and resource availability, an optimal framework can be designed for HS code classification in heavy equipment contexts.

2.4.5 Loss Function and Hyperparameter Tuning in Machine Learning Optimization

In machine learning, model optimization is driven by two key components: the loss function and hyperparameter tuning. Both elements work together to enhance predictive accuracy and generalization (Bergstra & Bengio, 2012). A loss function quantifies the difference between the predicted and actual values, guiding model optimization. During training, the goal is to minimize the loss function to improve model accuracy.

Hyperparameter tuning involves systematically adjusting parameters to achieve the best model performance based on a defined evaluation metric. Techniques such as grid search, random search, and Bayesian optimization are commonly employed for this purpose. This process is critical for addressing challenges such as imbalanced datasets, high-dimensional feature spaces, and hierarchical relationships inherent to HS code classification tasks. Key hyperparameters in common algorithms:

1. Naive Bayes:
 - 1) Alpha (Smoothing Parameter): Prevents zero probabilities by applying Laplace smoothing. This is particularly useful for sparse textual data generated from TF-IDF vectorization. Higher alpha values reduce variance but may oversimplify predictions, affecting classification accuracy for nuanced data.

2. Decision Trees and Random Forest:

- 1) Number of Trees (n_estimators): Defines the number of decision trees in the ensemble. A higher number generally improves performance but increases computational cost.
- 2) Maximum Depth (max_depth): Limits the depth of each tree to control overfitting while maintaining sufficient model complexity.
- 3) Minimum Samples Split (min_samples_split): Ensures each split in the tree contains sufficient data points, reducing noise in predictions.

3. K-Nearest Neighbors (KNN):

- 1) Number of Neighbors (k): Determines the number of nearest neighbors considered during classification. Higher values promote generalization but may overlook local patterns in the data.
- 2) Distance Metric: Defines how proximity between data points is calculated, with common metrics including Euclidean and Manhattan distances.

4. Linear Support Vector Machines (SVM):

- 1) Regularization Parameter (C): Controls the trade-off between achieving a low error rate on the training data and maximizing the margin between classes.
- 2) Kernel Function: Determines the transformation of input data into higher dimensions for better separability.

5. AdaBoost:

- 1) Learning Rate: Regulates the contribution of each weak learner in the ensemble. Lower values help avoid overfitting by slowing the adaptation to data.
- 2) Number of Estimators (n_estimators): Indicates the number of weak classifiers combined, balancing model complexity and computational efficiency.

If hyperparameters are not properly tuned, models can either overfit (memorizing training data rather than generalizing) or underfit (failing to capture meaningful patterns) (Bergstra & Bengio, 2012).

GridSearchCV: Systematic Hyperparameter Optimization

GridSearchCV is a widely used method for hyperparameter tuning that exhaustively searches over a specified parameter grid to determine the optimal combination for a

given model. It operates by evaluating all possible combinations of hyperparameters and selecting the set that yields the best performance based on a predefined scoring metric. GridSearchCV is implemented in Python using the *sklearn.model_selection.GridSearchCV* module.

Process of GridSearchCV:

1. Define the Parameter Grid: Specify the hyperparameters and the range of values to be tested.
2. Model Training and Evaluation: For each combination, train the model and evaluate its performance using cross-validation to ensure robustness.
3. Select the Best Combination: Identify the hyperparameter set that achieves the highest performance metric.

While GridSearchCV does not directly minimize the loss function, it seeks hyperparameter configurations that lead to models with optimal performance, thereby indirectly contributing to loss minimization.

Hyperparameter tuning significantly impacts the performance of HS code classification models. Random Forest, with optimized parameters, demonstrated the best balance between accuracy, interpretability, and computational efficiency. Future research could explore Bayesian Optimization further to reduce computational costs while maintaining robust performance. Additionally, implementing adaptive hyperparameter tuning methods could further refine classification accuracy in real-time applications.

2.4.6 Evaluation Methods in HS Code Classification ML

The studies reviewed employed various evaluation methods tailored to the challenges of HS code classification, focusing on accuracy, F1-score, and cross-validation. Key findings include:

1. Altaheri & Shaalan (2020): Accuracy was used to evaluate SVM and Naive Bayes models, which performed well on simple descriptions but struggled with hierarchical HS codes. Cross-validation was mentioned but not elaborated, limiting robustness for complex classifications.

2. Muslim (2022): The Naive Bayes model achieved 99.97% accuracy and a Kappa coefficient of 0.999 with structured data. However, it faced challenges with ambiguous and incomplete product descriptions, highlighting the limitations of real-world data.
3. Paramartha, Ardiyanto and Hidayat, (2021): Random Forest and Multinomial Naive Bayes models were evaluated using F1-score, precision, and recall, with k-fold cross-validation ensuring reliability. While effective on clean, multilingual data, the models struggled with unclear descriptions.
4. Shubham (2023): An ensemble approach improved scalability using precision-recall curves and hierarchical evaluation metrics. However, performance declined at granular classification levels, particularly for underrepresented classes.
5. Chen, et al., (2021): An unsupervised learning model with sentence embeddings achieved 71% accuracy for Dutch customs data. Despite promising results, challenges included limited labeled data and adapting across national HS code systems.
6. Du et al., (2021): HScodeNet, a hierarchical model, demonstrated improved performance using hierarchical relationships in HS codes. However, scalability issues arose due to dynamic updates in the HS code system.

To ensure the best possible performance of the model, it is important to choose the right evaluation metrics that align with the classification goals, particularly given the imbalanced datasets and the hierarchical nature of HS code classification:

1. Accuracy

Accuracy is commonly used but may not be sufficient in cases of class imbalance. In HS code classification, certain parts or categories are underrepresented, so accuracy alone may not reflect the true performance of the model.

2. F1-Score

F1-score is a crucial metric for balancing precision (the correctness of positive predictions) and recall (the ability of the model to find all relevant instances). In HS code classification, this is particularly important when dealing with

imbalanced data, as it avoids favoring the majority class while penalizing false positives and false negatives.

3. Precision-Recall Curve

The precision-recall curve provides more detailed insights into how well the model performs across different thresholds, which is important when the dataset has a high-class imbalance.

4. Confusion Matrix

A confusion matrix can be used to visualize the performance of the model in a more granular way, helping to identify misclassifications and providing insights into which specific categories or parts are being misclassified.

2.4.7 Related Work HS Code Classification Using Machine Learning

Machine learning (ML) has been extensively applied to automate HS code classification due to its ability to process large amounts of product data quickly. However, traditional ML models, such as Support Vector Machines (SVM), Naive Bayes, and Decision Trees, still face significant challenges in handling hierarchical data, ambiguous product descriptions, and the dynamic nature of global trade regulations. Altaheri & Shaalan (2020) showed that models like SVM and Naive Bayes perform well for basic product descriptions but struggle with complex, hierarchical HS code structures. They lack the ability to handle the relationships between product categories, leading to incorrect classifications in more intricate cases.

Muslim (2022) applied the Naive Bayes algorithm to Indonesian customs data, achieving high accuracy (99.97%) for structured data. However, the model struggled with incomplete or ambiguous product descriptions, highlighting its limitations in handling real-world, messy datasets. Paramartha, Ardiyanto and Hidayat, (2021) Similarly, evaluated Random Forest and Multinomial Naive Bayes models. While effective with clean, multilingual data, these models faltered when descriptions were inconsistent, underscoring their reliance on extensive data preprocessing, which is often impractical in real-world scenarios.

Du et al., (2021) proposed HScoreNet, a hierarchical model that improved classification by recognizing relationships between product categories. Although it enhanced accuracy, the model lacked flexibility, struggling to adapt to dynamic updates in HS codes and scale effectively. Shubham, et al., (2023) advanced this by combining transformer models like BERT with Knowledge Graphs in an ensemble approach, improving scalability and context handling. However, challenges persisted at the granular level of HS code hierarchies, indicating limitations in fine-grained classification even with sophisticated ensemble models.

These studies emphasize the need for models capable of handling incomplete, inconsistent data while maintaining flexibility and scalability to adapt to dynamic trade environments. Chen et al., (2021) used unsupervised learning with sentence embeddings to predict HS code correctness. Although this approach showed promise, it was hindered by the lack of labelled data and struggled to adapt across different countries' HS code systems.

Table 2.6 Related Works of HS Code Machine Learning

Related Work	Methods	Findings	Research Gap
Altaheri & Shaalan (2020)	Support Vector Machines, Decision Trees, Naïve Bayes	ML models perform well but struggle with complex and hierarchical data; need for contextual understanding	Lack of handling hierarchical structures and complex product descriptions
Muslim (2022)	Naïve Bayes Algorithm	Naïve bayes achieves high accuracy (99.97%) in structured data but struggles with ambiguous/incomplete descriptions	Ineffective when dealing with incomplete or ambiguous data
Paramartha, Ardiyanto and Hidayat, (2021)	Random Forest, Multinomial Naïve Bayes	Random Forest handles large datasets, but misclassifications occur without proper preprocessing	Challenges in multilingual and inconsistent product data handling
Du, Wan and Lin, (2021)	HScoreNet (Hierarchical model)	Hierarchical model improves classifications accuracy, especially for	Scalability challenges and issues in adapting of

Related Work	Methods	Findings	Research Gap
		hierarchical data structure like HS codes	dynamic updates of HS codes
Subhan (2023)	Ensemble-based Approach (Bert-Transformer, NER, Knowledge Graph)	The ensemble model successfully addresses scalability and automation in HS code classification	Difficulties remain in capturing nuances at granular levels of classification
Chen, Wang and Jin, (2021)	Unsupervised ML With Sentence Embedding	Achieves 71% accuracy in HS code correctness prediction using unsupervised methods	Limited labelled datasets for training and challenges in cross-border adaptation

The table 2.6 of related works highlights several key trends and research gaps in traditional ML models for HS code classification:

1. **Complexity of Hierarchical Data:** Most of the models reviewed, such as those from Altaheri & Shaalan (2020) and (Du et al., 2021), struggle with the hierarchical structure of HS codes. Traditional models like Naive Bayes and SVM lack the ability to account for relationships between nested categories, making them ineffective for complex classifications.
2. **Ambiguity and Incompleteness in Product Descriptions:** Models like Naive Bayes and Random Forest, as shown by Muslim (2022) and Paramartha, Ardiyanto and Hidayat, (2021), perform well with structured data but struggle when product descriptions are ambiguous or incomplete. This is a significant limitation, as real-world product data is rarely clean and often includes vague descriptions.
3. **Scalability and Dynamic Updates:** Studies like Du et al., (2021) and Shubham (2023) emphasize the challenges of scaling models to handle large datasets and dynamic updates in HS codes. While these models can automate classification, they fail to adapt in real-time to changes in trade regulations or updates in product categories, a critical requirement for modern customs and trade environments.

2.4.8 Research Gap of Machine Learning in HS Code Classification Automation

The complexity of HS code classification presents significant challenges that prevent it from being fully automated. While machine learning (ML) enhances efficiency, it cannot completely replace human expertise due to the nuanced, hierarchical, and evolving nature of HS codes. Instead, a decision support system (DSS) that integrates ML with human oversight is necessary to speed up the classification process while ensuring accuracy, compliance, and traceability.

Strengths of Machine Learning in HS Code Classification

ML significantly improves classification efficiency, reducing the manual workload required for processing large volumes of product descriptions. Studies such as Muslim (2022) and Altaheri & Shaalan (2020) highlight how Naïve Bayes, SVM, and Decision Trees achieve high accuracy with structured data, automating repetitive classification tasks and minimizing errors caused by human oversight. Additionally, ML models can identify patterns within textual data, providing faster classification recommendations to support customs officers and trade compliance teams.

Limitations of Fully Automated HS Code Classification

Despite its strengths, ML alone is not sufficient to handle the complexities of HS code classification, which requires contextual interpretation, regulatory adaptation, and transparency.

1 Context and Ambiguity in Product Descriptions

HS code classification is highly dependent on product context, which ML models often misinterpret. Muslim (2022) showed that Naïve Bayes performs well with structured customs data but struggles with ambiguous or incomplete product descriptions. ML lacks the ability to reason beyond predefined patterns, making human oversight essential for interpreting nuanced classifications.

2 Hierarchical Complexity and Regulatory Adaptation

HS codes are hierarchically structured and frequently updated, making static ML models ineffective over time. Research by Altaheri & Shaalan (2020) and Du et al. (2021) found that while hierarchical models such as HScodeNet improve classification accuracy, they fail to adapt to regulatory changes without frequent

retraining. A decision support system must allow human intervention to refine classifications and incorporate evolving regulations dynamically.

3 Traceability and Compliance Requirements

ML models often operate as black-box systems, meaning they provide predictions without justification. Regulatory audits and trade compliance require clear, explainable classification decisions. Shubham et al. (2023) emphasized the need for traceability, as customs officials must be able to track and justify classification choices. Without transparency, users are hesitant to rely on automated systems, reducing adoption and efficiency.

4 Decision Support System: Balancing Automation and Human Oversight

Instead of full automation, a decision support system (DSS) is the optimal approach leveraging ML for efficiency while maintaining human judgment for accuracy and regulatory compliance. This hybrid system would:

- Speed up classification by providing automated recommendations based on learned patterns.
- Ensure traceability by allowing users to review, modify, and justify classifications.
- Adapt to changing regulations with human oversight guiding updates and refinements.

5 Operational Risks

- 1) Misclassification: Errors result in financial penalties, customs delays, and increased administrative efforts.
- 2) User Distrust: Without transparency, users are hesitant to rely on automated systems, reducing adoption and efficiency.
- 3) Scalability: Real-time, large-scale applications face computational challenges, compounded by the black-box nature of predictions.

In conclusion HS code classification cannot be fully automated due to its hierarchical structure, evolving regulations, reliance on tacit knowledge, and the ambiguity of product descriptions. These complexities make it challenging for traditional machine learning (ML) models to independently provide accurate and consistent classifications.

The need for human expertise in interpreting regulatory changes and handling edge cases remains critical.

One promising approach to enhancing context awareness and hierarchical understanding is the use of knowledge graphs (KGs). Unlike traditional ML models, KGs can capture relationships between product categories, allowing for more structured and explainable classifications. The next section explores how knowledge graphs can complement ML by incorporating domain knowledge, improving traceability, and adapting to regulatory changes, strengthening the decision support framework for HS code classification.

2.5 Knowledge Graph in HS Code Classification

2.5.1 Knowledge Graph in Hierarchical System

Knowledge Graphs (KGs) provide a robust framework for managing hierarchical systems by structuring and reasoning over complex relationships. Their applications in medical coding, regulatory compliance, and product classification demonstrate their utility, offering insights for HS code classification. However, challenges such as resource-intensive construction, scalability, and integration of dynamic updates need to be addressed to fully leverage KGs in this domain. Applications of KGs in Hierarchical Systems:

1. Medical Coding and Diagnosis

Rajabi and Kafaie (2022) utilized KGs to manage the International Classification of Diseases (ICD), encoding relationships among symptoms, diagnoses, and treatments. This hierarchical structure improved diagnostic accuracy and consistency, emphasizing interoperability and decision support. Similar methodologies can be applied to HS codes, linking part numbers and descriptions to standardized classifications. Challenges noted include the resource-intensive nature of KG construction and the need for continuous updates.

2. Regulatory Compliance

Xi (2024) demonstrated KGs' ability to integrate legal frameworks into financial systems, enabling real-time compliance monitoring and traceability. For HS code classification, KGs could map trade regulations like BTKI into hierarchical

frameworks. However, scalability and adaptability to dynamic regulations remain challenges, especially in large-scale implementations.

3. Supply Chain and Product Classification

(Peng et al., 2023) applied KGs to supply chain optimization, linking materials, components, and products in hierarchical taxonomies. This approach aligns with HS code classification by facilitating reasoning about product attributes and their categories. Despite the advantages, the study highlighted the significant resources required for constructing and maintaining such KGs in dynamic domains.

4. KG Construction for Hierarchical Systems

Yuan, et al., (2019) proposed a semi-supervised KG construction approach for unstructured biomedical texts, reducing manual effort. This technique is relevant for structuring unstructured regulatory texts like BTKI into KGs for HS code reasoning. However, accurate entity and relationship extraction methods are critical and remain a challenge.

Insights for HS Code Classification:

1. **Structural Alignment:** KGs can model hierarchical relationships between part numbers, product attributes, and HS codes, similar to their use in ICD codes and supply chains.
2. **Regulatory Integration:** Real-time integration of regulations like BTKI within a KG framework enhances adaptability and compliance.
3. **Automation Potential:** Semi-supervised methods for KG construction reduce manual effort but require reliable entity extraction techniques.
4. **Scalability and Maintenance:** Dynamic domains like global trade demand scalable KG systems capable of handling frequent updates.

Table 2.7 Related Work Application on Knowledge Graph and LLM

Domain	Study	Methodology	Findings/Contributions	Challenges/Limitations
Medical Coding	Rajabi and Kafaie (2022)	Hierarchical KG for ICD and SNOMED classification	Enhanced diagnostic accuracy and interoperability through structured reasoning over medical data.	Dependency on domain-specific ontologies; frequent updates required for accuracy.
Regulatory Compliance	Xi (2024)	Financial KG mapping regulations to transactions	Improved compliance monitoring by linking hierarchical legal frameworks to financial data.	Maintaining up-to-date regulations and high computational overhead for large-scale KGs.
Supply Chain	Peng et al., (2023)	KG integration for supply chain optimization	Identified bottlenecks and streamlined logistics using hierarchical relationships among materials and components.	Scalability challenges in rapidly changing supply chain environments.
KG Construction	Yuan, et al., (2019)	Semi-supervised KG construction from unstructured data	Demonstrated effective entity and relationship extraction for biomedical KGs with minimal supervision.	Quality heavily depends on the accuracy of extraction methods and integration of noisy data.
Product Classification	Peng et al., (2023)	Reasoning over hierarchical product taxonomies	Enhanced classification consistency by linking product attributes to hierarchical categories.	Resource-intensive graph construction; difficulties in handling diverse product attributes.
Healthcare Decision Support	Teng et al., (2020)	Medical KG based on EMR integration	Improved clinical decision-making and diagnosis clustering through embedding-based graph reasoning.	Complexity in handling unstructured EMRs and dynamic medical standards.

Table 2.7 summarizes related works exploring their applications and innovations. Knowledge graphs (KGs) offer robust capabilities for addressing the hierarchical complexity of HS code classification by representing relationships between part

numbers (PNs), product attributes, and trade regulations in a structured and semantically rich format. They enable semantic reasoning, ensuring consistent and accurate classification while reducing errors. By embedding regulatory frameworks like BTKI as ontological rules, KGs facilitate automated compliance checks, aligning classifications with evolving regulations. Furthermore, KGs enhance transparency by providing clear reasoning paths for decisions, making them ideal for audit-ready processes. Opportunities lie in automating classification through integration with machine learning models, where KG embeddings improve predictive accuracy and scalability. Dynamic update mechanisms ensure adaptability to regulatory changes, while systematic mapping of PNs to HS codes enhances efficiency, particularly in industries with complex product taxonomies like heavy equipment manufacturing. Cross-domain applications, such as linking supply chain data with trade regulations, and hybrid systems combining KGs with large language models (LLMs) further expand their transformative potential, offering scalable, accurate, and explainable solutions for modern classification challenges.

Despite the demonstrated effectiveness of knowledge graphs (KGs) in hierarchical systems, several challenges and limitations hinder their application to HS code classification:

1. **Resource-Intensive Construction:** Building and maintaining domain-specific KGs require significant manual effort, particularly in data-heavy fields like healthcare and product classification (Rajabi and Kafaie, 2022; Yuan, et al., 2019).
2. **Scalability Issues:** Large-scale systems with evolving hierarchies, such as HS codes and regulatory frameworks, face computational and organizational scalability challenges (Peng et al., 2023; Xi, 2024).
3. **Dynamic Updates:** KGs struggle with incorporating frequent updates, such as those required for regulatory frameworks like BTKI, limiting their adaptability to changing trade regulations (Xi, 2024).

4. **Explainability Constraints:** While KGs are more interpretable than many AI models, providing clear, human-readable reasoning paths for classification decisions remains an ongoing challenge (Rajabi and Kafaie, 2022).
5. **Integration Gaps:** Limited research exists on effectively linking product-specific part numbers and attributes to hierarchical systems like HS codes or integrating trade regulations into a unified KG framework.

These challenges highlight the need for innovative approaches to improve scalability, dynamic adaptability, and integration while ensuring transparent and efficient HS code classification.

2.5.2 Introduction to Knowledge Graph

Knowledge Graphs (KGs) are structured representations of information that depict entities and their interrelations within a graph framework. Each node signifies an entity, and each edge represents a relationship between entities, enriched with semantic context to facilitate machine understanding and reasoning. This structure enables the integration of diverse data sources, supporting applications in artificial intelligence and data analytics (Hogan et al., 2021).

Key characteristics of Knowledge Graphs include:

1. **Graph-Based Representation:** Entities are depicted as nodes, and their relationships are represented as edges, forming a network that mirrors real-world connections.
2. **Triplet Structure:** KGs often use (subject, predicate, object) triples, such as (Track Assembly, is Component Of, Bulldozer), to encode relationships in a form that is both human-readable and machine-processable.
3. **Semantic Enrichment:** KGs incorporate ontologies and metadata to provide context, enabling systems to accurately interpret the meaning of entities and their relationships.
4. **Flexibility and Scalability:** The graph structure allows for the seamless addition of new entities and relationships without necessitating a complete schema redesign, accommodating the dynamic nature of real-world data.

5. Support for Reasoning and Inference: KGs facilitate logical reasoning, deriving new insights from existing data through exploring relationships and hierarchies.

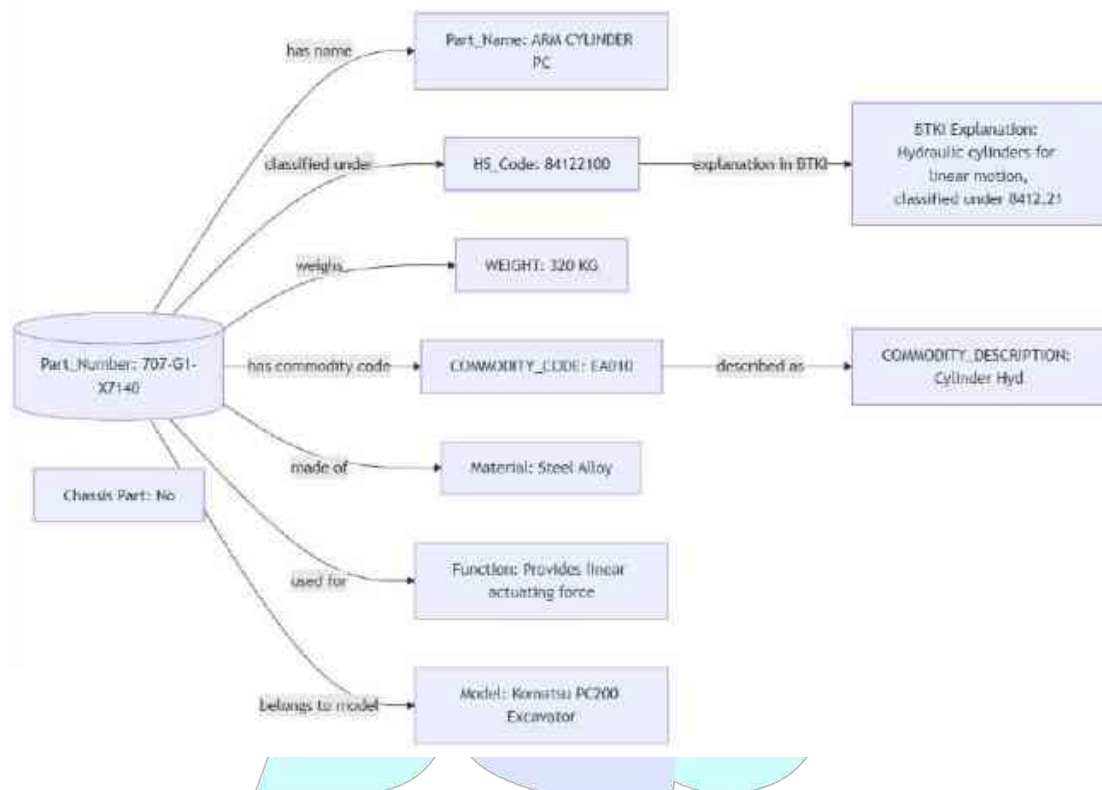


Figure 2.11 Example of a Knowledge Graph Applied to HS Code Classification in the Heavy Equipment Industry.

This diagram in Figure 2.11 illustrates a simplified Knowledge Graph for a hydraulic cylinder part, where nodes represent entities such as the Part Number (707-G1-X7140), its attributes (e.g., Part Name, Weight, Material, etc.), and the HS Code (84122100) along with its BTKI explanation. Edges denote relationships like “has name”, “classified under”, and “belongs to model”. The graph demonstrates how various aspects of the part—its function, commodity details, and material composition are interconnected and classified within the HS Code system. By capturing these relationships in a Knowledge Graph, this example highlights how complex data (such as product attributes, usage context, and trade classification) can be organized, related, and managed to ensure accurate classification and efficient data integration across heavy equipment inventories.

In the context of HS (Harmonized System) Code classification, particularly within the heavy equipment industry, KGs offer significant advantages:

1. **Enhanced Data Integration:** KGs consolidate information from various sources, such as product specifications, supplier details, and regulatory requirements, into a unified framework. This integration streamlines the classification process by providing a comprehensive view of all relevant data (Chaudhri et al., 2022).
2. **Improved Accuracy in Classification:** By modelling complex relationships and hierarchies inherent in HS Codes, KGs enable more precise classification of products. The semantic richness of KGs allows for differentiation between similar items based on nuanced attributes, reducing errors in classification (Pan et al., 2023).
3. **Adaptability to Regulatory Changes:** The flexible nature of KGs ensures that updates to regulations or classification criteria can be incorporated efficiently, maintaining the relevance and accuracy of the classification system over time (Chaudhri et al., 2022).
4. **Facilitation of Automated Reasoning:** KGs support automated reasoning capabilities, enabling the system to infer appropriate HS Codes for new or modified products based on existing knowledge, thereby enhancing operational efficiency (Pan et al., 2023).

2.5.3 Historical Context and Development

The concept of Knowledge Graphs (KGs) can be traced back to foundational developments in graph theory and semantic web technologies. Tools such as the Resource Description Framework (RDF) and ontologies laid the groundwork for structuring relationships between entities, enabling interoperability and reasoning over distributed datasets (Hogan et al., 2021). For example, RDF models represent data as triples subject, predicate, and object allowing semantic representation of relationships like (Component, is Used In, Excavator).

The term "Knowledge Graph" gained prominence in 2012 with Google's launch of its Knowledge Graph to enhance search capabilities. This marked a significant milestone,

demonstrating the scalability and utility of KGs in integrating and retrieving vast datasets (Hogan et al., 2021). Subsequently, KGs were adopted across various domains:

1. **Healthcare:** KGs model relationships among symptoms, diseases, and treatments to enhance diagnosis and decision-making processes. For instance, SNOMED CT and ICD systems utilize KGs to provide structured reasoning over hierarchical medical codes (Rajabi and Kafaie, 2022).
2. **Supply Chains:** KGs facilitate the mapping of materials, components, and products to optimize logistics and inventory management. This includes applications in identifying bottlenecks and improving the traceability of goods across global supply chains (Peng et al., 2023).
3. **Regulatory Compliance:** Integration of legal frameworks into KGs enables real-time monitoring and adherence to regulations in industries like finance and trade. For example, financial compliance systems often use KGs to model regulatory requirements and automate reporting processes (Xi, 2024).

In recent years, significant advancements have expanded the capabilities of KGs, making them indispensable for complex domains like HS Code classification. Key developments include:

1. **Graph Embeddings and Machine Learning Integration:** Techniques such as graph embeddings allow KGs to encode relationships in a form suitable for machine learning models. This integration has enabled predictive modeling in areas such as component compatibility and supply chain risk analysis (Pan et al., 2023).
2. **Dynamic and Real-Time Graphs:** Modern KGs can handle dynamic updates and real-time data streams, making them ideal for tracking heavy equipment parts and their associated regulations. This adaptability supports the continuous alignment of classification systems with evolving trade standards (Hogan et al., 2021).
3. **Interoperability with Standards:** KGs have demonstrated interoperability with classification systems such as HS Codes, ensuring seamless integration of trade regulations and enabling automated compliance checks ((Chaudhri et al., 2022).
4. **Explainability and Transparency:** By visualizing relationships and reasoning paths, KGs improve the interpretability of automated systems. For instance, trade

compliance platforms can illustrate the decision path for assigning a specific HS Code to a product, enhancing trust and reducing errors (Pan et al., 2023).

Understanding the historical development and advancements of Knowledge Graphs provides a solid foundation for appreciating their transformative potential. The following section will delve into the rationale behind adopting Knowledge Graphs for HS Code classification, demonstrating how they address the limitations of traditional systems and enhance semantic reasoning and hierarchical data management.

2.5.4 Building the Knowledge Graph

A Knowledge Graph (KG) for HS code classification is a structured representation of interconnected data entities and their relationships. The graph organizes complex data such as part descriptions, HS codes, supplier information, and regulatory requirements into a network of nodes (entities) and edges (relationships). This design facilitates intuitive querying, reasoning, and classification for parts in a hierarchical and relational framework.

The primary purpose of the KG is to capture the rich contextual information required to classify heavy equipment parts into appropriate HS codes accurately. This context includes details about the parts' material composition, functionality, suppliers, and applicable trade regulations. Key Components of the Knowledge Graph consist of:

1. Entity Modelling (Nodes): Representing key entities such as parts, HS codes, suppliers, and regulations.
 - 1) Part Descriptions: The central focus, representing unique components in the system. For instance, a part number (PN) like *707-G1-X7140* has associated details such as its name (*Arm Cylinder*), material, and functional description.
 - 2) HS Codes: Nodes that correspond to the Harmonized System classifications, encoding hierarchical data from broad categories to specific subcategories.
 - 3) Suppliers and Commodities: Nodes that capture additional dimensions of the parts, such as their origin (supplier) and groupings based on material or function (commodity groups).

- 4) Regulatory Requirements: Nodes representing trade policies, such as required documentation, duties, and restrictions associated with specific HS codes.
2. Relationships (Edges): Representing the relationships between these entities, such as a part being classified under an HS code or a supplier providing specific parts.
 - 1) Classification Relationships: Linking parts to their HS codes (e.g., *Part Number 707-G1-X7140* → *classified_as* → *HS Code 84122100*).
 - 2) Product Information Associations: Connecting parts to suppliers (e.g., *Part Number 707-G1-X7140* → *weight* → *200,000 gram*).
 - 3) Hierarchical Relationships: Structuring HS codes in their natural hierarchy (e.g., *HS Code 84122100* → *part_of* → *Heading 8412* → *part_of* → *Chapter 84*).
 - 4) Regulatory Connections: Associating HS codes with compliance and trade policies (e.g., *HS Code 84122100* → *regulated_by* → *BTKI Regulation*).

Once constructed, the Knowledge Graph's operational workflows define how it is utilized and maintained. These workflows are crucial to ensuring its relevance and effectiveness in dynamic domains (Pan et al., 2023)..

1. Querying and Reasoning:
 - 1) Allow users to query the KG for specific parts or classifications (e.g., "Which parts fall under HS Code 8412.2100?").
 - 2) Enable reasoning engines to infer new relationships or validate classifications.
2. Updates and Maintenance:
 - 1) Regularly update the KG to reflect changes in regulations, tariffs, or product information.
 - 2) Use dynamic graph tools to ensure seamless integration of new data.

Benefits of a Well-Structured Knowledge Graph:

- 1) Enhanced Efficiency: Streamlines HS Code classification and reduces manual effort.
- 2) Improved Accuracy: Ensures precise and consistent classifications aligned with regulations.
- 3) Adaptability: Easily incorporates updates and new regulations.

While Knowledge Graphs have proven transformative, their adoption comes with challenges:

1. **Resource-Intensive Construction:** Building and maintaining domain-specific KGs require significant manual effort, especially in fields with complex datasets.
2. **Computational Overhead:** Dynamic updates and real-time reasoning increase computational demands, necessitating scalable infrastructure.
3. **Data Integration:** Integrating unstructured or noisy data into KGs remains a technical hurdle, impacting the completeness and accuracy of the graphs.
4. **Adaptability to Rapid Changes:** Frequent updates in domains like trade regulations require efficient mechanisms for modifying KGs without compromising consistency.

The field of Knowledge Graphs continues to evolve with advancements that expand their capabilities and applications. Below are three key trends and innovations that hold promise for the future of Knowledge Graphs in HS Code classification and beyond (Hogan, et al., 2021; Pan, et al., 2023; Xi, 2024):

1. **Integration with Large Language Models (LLMs):**

The integration of KGs with LLMs, such as GPT, has enabled enhanced contextual reasoning and retrieval-augmented generation. This synergy allows for more accurate and nuanced responses to complex queries by leveraging the structured knowledge in KGs and the language understanding of LLMs (Pan et al., 2023).

2. **Real-Time Knowledge Graphs:**

Advances in dynamic and real-time graph technologies enable Knowledge Graphs to handle continuous data updates seamlessly. These innovations are particularly valuable in trade compliance, where regulations and tariffs frequently change, ensuring that the KG remains current and reliable (Rajabi and Kafaie, 2022; Xi, 2024).

3. **Graph Embeddings for Predictive Analytics:**

The use of graph embeddings, which transform graph structures into numerical representations, has opened new possibilities in predictive analytics. For example,

these embeddings can identify patterns in HS Code classifications or predict the compatibility of new products with existing codes, streamlining decision-making (Peng et al., 2023).

The advancements and emerging trends in Knowledge Graphs showcase their transformative potential in handling complex and dynamic datasets. Having established a comprehensive understanding of their construction and innovations, the next section will delve into their practical applications.

2.5.5 Rationale of Knowledge Graphs

The rationale for adopting Knowledge Graphs (KGs) in domains like HS Code classification lies in their ability to overcome the limitations of traditional data management systems, such as relational databases. Unlike conventional systems, KGs provide a flexible, semantic-rich framework that facilitates efficient data integration, hierarchical reasoning, and dynamic adaptability.

While relational databases have been the cornerstone of data management for decades, their rigid structure often falls short in handling the complexities of semantic relationships and hierarchical data. Studies by Kotiranta et al. (2022) and Ramosaj et al. (2023), emphasize the distinctions between relational databases and knowledge graphs, highlighting the latter's advantages in semantic representation, reasoning capabilities, and adaptability. Table 2.8 outlines these key differences:

Table 2.8 Comparison of Relational Databases and Knowledge Graphs

Aspect	Relational Databases	Knowledge Graphs
Semantic Representation	Stores relationships as foreign keys, lacks semantic context.	Enriches data with semantic annotations and ontologies, enabling contextual reasoning.
Hierarchical Structures	Requires complex joins and recursive queries to represent hierarchies.	Encodes hierarchies as directed edges, enabling efficient traversal and updates.
Reasoning and Automation	No native reasoning capabilities; external logic is required.	Supports automated reasoning to infer

Aspect	Relational Databases	Knowledge Graphs
		relationships and validate classifications.
Dynamic Adaptability	Schema changes are required for updates, risking workflow disruptions.	Schema-flexible; new nodes and edges can be added without redesign or disruption.
Use Case Suitability	Best for static datasets with simple relationships.	Ideal for dynamic, interconnected datasets requiring semantic depth and hierarchy.

Key Advantages for HS Code Classification:

1. **Semantic Representation:** Relational databases cannot encode semantic meaning directly, making them unsuitable for nuanced classifications. KGs enrich data with semantic annotations, capturing domain-specific rules like "Hydraulic Pump is Used In Construction Machinery" and linking it to "HS Code 8412.2100," which specifies subcategories at the eight-digit level (Hogan et al, 2021). This capability enables automated reasoning, reducing manual intervention.
2. **Hierarchical Representation:** The hierarchical nature of HS Codes—from chapters to subheadings—is cumbersome to model in relational databases. KGs efficiently represent such structures using directed edges, ensuring intuitive traversal and seamless updates. For example, the classification tree for "8412.2100" (parts of hydraulic power engines) links broader categories like "8412" (Hydraulic Power Engines) to more specific categories (Chaudhri et al., 2022).
3. **Reasoning and Automation:** Relational databases require external systems to infer classifications. In contrast, KGs leverage reasoning engines to deduce relationships. For example, if "Hydraulic Pump is Used In Excavator" and "Excavator is Classified As HS Code 8412.2100," a KG can infer "Hydraulic Pump is Classified As HS Code 8412.2100" (Pan et al., 2023).
4. **Dynamic Adaptability:** Frequent updates to HS Codes or regulations demand flexibility. Relational databases face challenges with schema redesigns, while

KGs seamlessly integrate new rules and entities, such as linking "HS Code 8412.2100" to "Local Tariff 5%".

This diagram in Figure 2.12 demonstrates how KGs represent hierarchical relationships and product classifications. Nodes depict entities like products and HS Codes, while edges illustrate their relationships, such as "is used in" and "classified as."

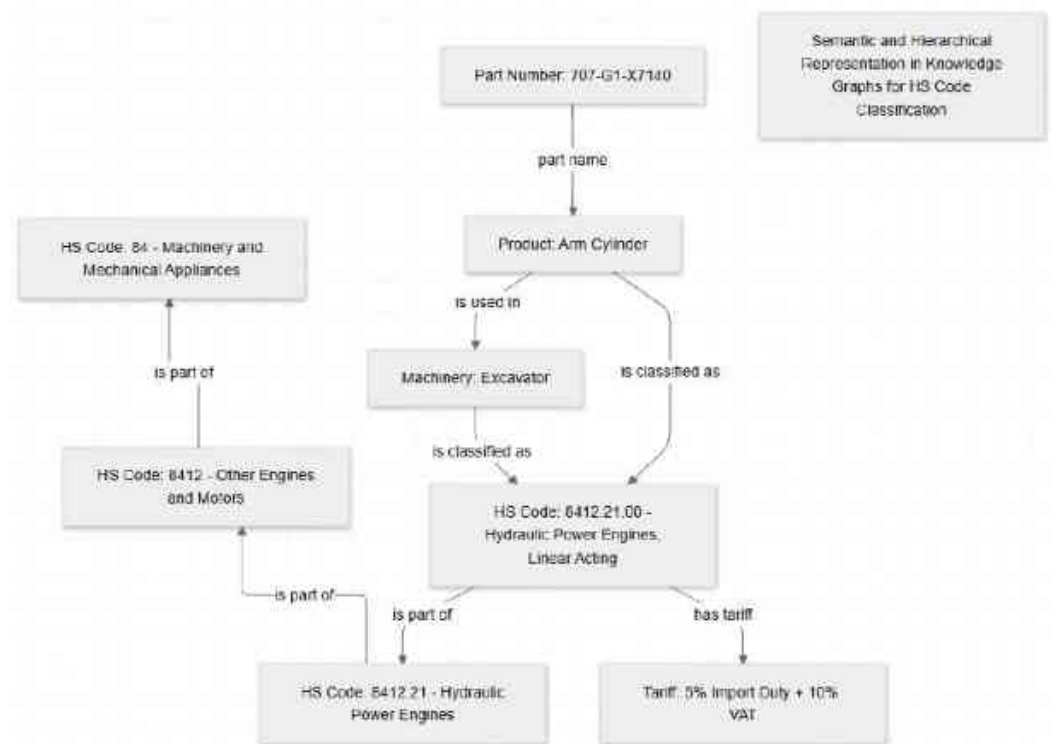


Figure 2.12 Semantic and Hierarchical Representation in Knowledge Graphs for HS Code Classification

Challenges Addressed by Knowledge Graphs:

- 1) Complexity of Regulations: KGs capture and apply intricate classification rules and exceptions through semantic enrichment.
- 2) Interoperability: KGs integrate seamlessly with trade compliance platforms and regulatory databases, ensuring consistency.
- 3) Improved Decision-Making: By visualizing reasoning paths, KGs enhance transparency and confidence in classification decisions.

Knowledge Graphs significantly improve HS Code classification by offering semantic depth, hierarchical modeling, adaptability.

2.6 Large Language Model (LLM) for Contextual Understanding

2.6.1 Introduction: What is a Large Language Model (LLM)?

Large Language Models (LLMs) are advanced artificial intelligence systems designed to process, understand, and generate human-like text. Built on neural network architectures like transformers, these models leverage vast datasets of structured and unstructured text to capture linguistic patterns and contextual relationships (Vaswani et al., 2017). LLMs excel in tasks like language translation, summarization, information retrieval, and domain-specific classification (Zhao et al., 2024). Their scalability and generalization capabilities make them indispensable across industries, including supply chain management and regulatory compliance (Larson and Truitt, 2024). LLMs rely on billions to trillions of parameters that encode relationships between words, phrases, and concepts. These parameters are learned through extensive pretraining on large text corpora and can be fine-tuned for specific applications (Yang et al., 2021). By employing advanced mechanisms like self-attention, LLMs achieve a deep understanding of text, enabling them to tackle complex problems in hierarchical and dynamic domains (Zhao et al., 2023).

2.6.2 Historical Overview of LLMs

The evolution of LLMs represents a series of milestones in artificial intelligence and natural language processing:

1. Rule-Based Systems (1950s–1980s)

Early models like ELIZA used rule-based heuristics to mimic conversation but lacked true understanding. These systems relied on predefined patterns and were limited to specific scripts (Yang et al., 2021).

2. Statistical Language Models (1990s–2000s)

Probabilistic approaches, such as n-grams and Hidden Markov Models, introduced statistical methods to predict word sequences. These models improved performance but struggled with capturing long-range dependencies and contextual relationships (Vaswani et al., 2017).

3. Neural Networks and Word Embeddings (2010s)

- 1) Word Embeddings: Techniques like Word2Vec and GloVe created dense vector representations of words, capturing semantic relationships more effectively (Zhao et al., 2024).
- 2) Recurrent Neural Networks (RNNs): These models modeled sequential data but were computationally intensive and prone to losing context over long sequences (Yang et al., 2021).
- 3) LSTM (Long Short-Term Memory): LSTM models addressed some of the limitations of RNNs by preserving long-range dependencies (Zhao *et al.*, 2024).

4. Transformer Architecture (2017)

A pivotal advancement came with Vaswani et al.'s introduction of the transformer model, which used self-attention mechanisms to process entire text sequences in parallel. This breakthrough became the foundation of modern LLMs (Vaswani et al., 2017).

5. Pretrained Models (2018–2019)

- 1) BERT (Bidirectional Encoder Representations from Transformers): Focused on bidirectional understanding of text, improving tasks like question answering and classification (Yang et al., 2021).
- 2) GPT (Generative Pretrained Transformer): Specialized in autoregressive text generation, setting new benchmarks in NLP performance (Larson and Truitt, 2024).

6. Modern LLMs (2020–Present)

Models like GPT-3, GPT-4, and PaLM scaled up to billions of parameters, exhibiting emergent capabilities like in-context learning and multi-step reasoning. Open-source efforts, such as BLOOM, and research-focused models like LLaMA further expanded the accessibility and applicability of LLMs (Zhao et al., 2024).

2.6.3 Types of Large Language Models (LLMs)

Large Language Models (LLMs) can be categorized into three primary types based on their architecture and functional design: encoder-only models, decoder-only models,

and encoder-decoder models. Each type serves specific purposes in natural language processing (NLP) tasks and offers unique advantages tailored to different applications:

1. **Encoder-Only Models:** BERT, RoBERTa.

Encoder-only models are optimized for understanding and encoding input text into dense, meaningful representations. These models focus on extracting information and identifying patterns within the input rather than generating new text. They are particularly effective for tasks requiring detailed comprehension of text, such as classification or entity recognition.

2. **Decoder-Only Models:** GPT Series, LLaMA.

Decoder-only models specialize in text generation, predicting the next word in a sequence based on previously generated words. These models are autoregressive, meaning they generate text one token at a time, using prior context to inform predictions.

3. **Encoder-Decoder Models:** T5, BART.

Encoder-decoder models combine the strengths of both encoder-only and decoder-only architectures. They first process input text into meaningful encoded representations (encoder) and then generate output text based on these representations (decoder). This design allows for simultaneous comprehension and generation, making these models versatile across tasks that require bidirectional understanding and output generation.

Table 2.9 Categorization of LLM and Comparative

Aspect	Encoder-Only Models	Decoder-Only Models	Encoder-Decoder Models
Focus	Understanding text	Generating text	Understanding and generating text
Examples	BERT, RoBERTa	GPT series, LLaMA	T5, BART
Applications	Sentiment analysis, classification	Content creation, conversational agents	Translation, summarization, QA
Strengths	Text comprehension	High-quality generation	Versatility across tasks

Aspect	Encoder-Only Models	Decoder-Only Models	Encoder-Decoder Models
Limitations	Cannot generate text	Prone to hallucination	Computationally intensive

Table 2.9 highlight categorization of LLMs into encoder-only, decoder-only, and encoder-decoder architectures highlights their diverse applications and strengths. Encoder-only models excel in tasks requiring comprehension, decoder-only models lead in text generation, and encoder-decoder models bridge both functionalities for complex applications. Understanding these distinctions is essential for selecting the appropriate model for specific tasks, whether in sentiment analysis, conversational AI, or translation.

2.6.4 LLMs Capabilities and Limitation

Modern Large Language Models (LLMs) have demonstrated exceptional abilities in understanding and generating text, enabling them to perform complex, domain-specific tasks with remarkable accuracy and efficiency. Their strengths lie in several key areas:

1. Semantic Understanding

LLMs leverage advanced transformer architectures, including self-attention mechanisms, to capture nuanced relationships between words, phrases, and sentences. This allows them to parse complex language constructs and derive meaningful insights from unstructured text. For example, they can identify product features, categorize items, and interpret regulatory requirements effectively, making them invaluable for tasks like product categorization and compliance management (Vaswani et al., 2017; Larson and Truitt, 2024).

2. Contextual Awareness

By generating embeddings, dense numerical representations of text, LLMs understand the broader narrative and context of terms within a given dataset. This contextual awareness enables them to disambiguate terms that may have multiple meanings. For instance, LLMs can distinguish between "pump" in the context of "hydraulic pump for construction" and "pump in medical equipment," providing precise, context-sensitive interpretations (Yang et al., 2021).

3. Reasoning and Inference

LLMs excel in multi-step reasoning, a capability that is particularly important for hierarchical tasks like HS code classification. They can infer relationships between various product attributes and regulatory standards, enabling them to navigate and classify complex datasets with high accuracy (Zhao et al., 2024).

4. Few-Shot and Zero-Shot Learning

Unlike traditional models that require extensive task-specific training, LLMs can generalize across new tasks with minimal (few-shot) or no (zero-shot) examples. This capability significantly reduces the need for large, domain-specific datasets, making them more adaptable to dynamic environments (Yang et al., 2021).

5. Retrieval-Augmented Generation (RAG)

By integrating with external knowledge bases, LLMs enhance their factual accuracy and relevance in generating outputs. This approach, known as Retrieval-Augmented Generation, allows LLMs to access and incorporate verified information, ensuring their outputs are not only accurate but also grounded in authoritative data (Larson and Truitt, 2024).

Despite their advanced capabilities, LLMs face notable limitations, particularly when applied to specialized domains like trade compliance and regulatory classification:

1) Dependence on Pre-Trained Data

LLMs are typically trained on general-purpose datasets that may not fully encompass the nuances of specialized fields. This can lead to inaccuracies when they encounter domain-specific terminology or evolving standards, such as changes in trade compliance regulations (Yang et al., 2021).

2) Limited Grasp of Regulatory Nuances

While LLMs can process hierarchical data, they often struggle with the complexities of domain-specific hierarchies. For example, differentiating between closely related HS codes or understanding exceptions in trade laws requires specialized knowledge that may not be present in their training data (Larson and Truitt, 2024).

3) Hallucination Risks

One of the critical challenges of LLMs is hallucination—generating outputs that appear plausible but are factually incorrect. This is particularly problematic in high-stakes applications like compliance and classification, where errors can have significant legal and financial repercussions (Zhao *et al.*, 2024).

4) Lack of Transparency

LLMs often operate as black-box systems, making it difficult to trace or explain their decision-making processes. This lack of transparency is a significant drawback in regulatory environments, where auditability and accountability are crucial (Yang *et al.*, 2021).

5) High Adaptation Costs

Adapting LLMs for specific domains through fine-tuning requires substantial computational resources and curated datasets. This process can be resource-intensive, limiting their scalability and accessibility in rapidly evolving industries like trade compliance (Larson and Truitt, 2024).

2.6.5 Retrieval-Augmented Generation (RAG) and LLMs

Retrieval-Augmented Generation (RAG) is a advanced framework designed to enhance the capabilities of Large Language Models (LLMs) by integrating external retrieval mechanisms. While LLMs such as GPT-4 and LLaMA exhibit remarkable generative and contextual reasoning abilities, they often rely on static pretraining datasets, which limit their ability to provide up-to-date or domain-specific information. This inherent limitation can result in the generation of hallucinated or factually inaccurate outputs (Lewis, *et al.* 2020)

By incorporating a retrieval component, RAG enables LLMs to dynamically access external information, allowing for contextually enriched and factually accurate responses. This innovation is particularly impactful for knowledge-intensive tasks, where factual precision and relevance are essential (Liu *et al.*, 2024).

The RAG framework operates through a structured workflow comprising three primary stages:

1. Query Generation:

The user's input is transformed into a retriever-compatible query by the LLM. This query reflects the context and intent of the user's request.

2. Data Retrieval:

The retriever module accesses external sources such as document repositories, databases, or APIs, extracting relevant information based on the generated query. Retrieval mechanisms often rely on vector-based search or semantic similarity measures.

3. Response Generation:

The retrieved content is incorporated into the LLM's input context, enabling it to generate a coherent response that combines retrieved factual data with its pre-trained language capabilities.

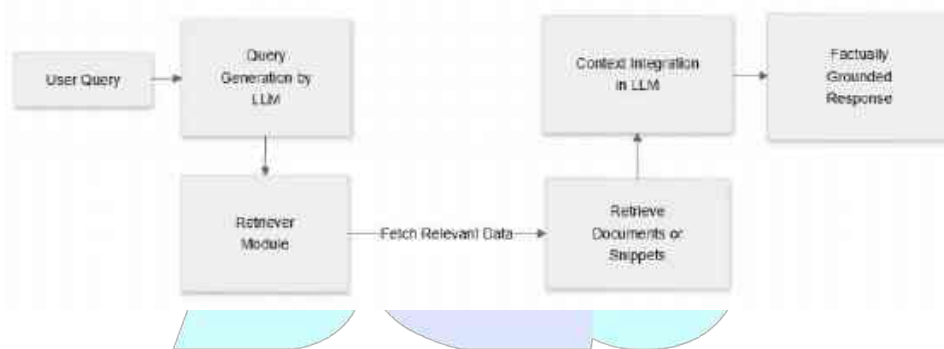


Figure 2.13 Workflow of a Retrieval-Augmented Generation (RAG) Framework.

This Figure 2.13 demonstrates the interaction between the user query, retrieval system, and Large Language Model (LLM). The retrieval system dynamically fetches external data to augment the LLM's response, ensuring factual accuracy and contextual relevance.

The success of a RAG framework heavily relies on the quality and diversity of its external data sources. These can be grouped into five main types:

1. Structured Data Repositories:

- 1) Relational Databases: Systems such as MySQL and PostgreSQL are used to store financial records, operational metrics, or compliance-related data (Xu, et al., 2024).

- 2) Spreadsheets: Tools like Excel and Google Sheets can hold tabular data relevant for analysis and decision-making.
 - 3) Knowledge Graphs: Represent structured and interconnected data for hierarchical reasoning tasks, often used in domain-specific contexts like regulatory or taxonomies (Jiang, Zhou, Zhao and Wen, 2023).
2. Unstructured Text Repositories:
- 1) Document Archives: These include research papers legal documents, and government publications (Lewis et al., 2020).
 - 2) Corporate Knowledge Bases: Internal organizational data, such as FAQs, standard operating procedures, or customer service logs (Xu et al., 2024).
 - 3) Web-Scraped Content: Forums, blogs, or open access repositories provide user-generated insights, useful for consumer sentiment analysis or trend detection (Liu et al., 2024).
3. Dynamic and Real-Time Sources:
- 1) Public APIs: These include tools like OpenWeatherMap[®] for real-time weather updates or Alpha Vantage for financial data (Lewis et al., 2020).
 - 2) Private APIs: Enterprise systems like Salesforce or custom supply chain management tools provide organizationally relevant, up-to-date data (Xu et al., 2024)
 - 3) News Feeds and Social Media: Aggregators such as RSS feeds or platforms like Twitter provide real-time data for current events or trend analysis.
4. Specialized and Domain-Specific Repositories:
- 1) Scientific Databases: Resources like PubMed and ClinicalTrials.gov provide highly specific data for medical or pharmaceutical research (Liu et al., 2024).
 - 2) Industry-Specific Directories: Trade compliance tools, such as HS code directories, or engineering manuals for technical operations (Xu et al., 2024).
5. Multimedia and Pretrained Models:
- 1) Multimedia Repositories: Platforms storing audio, video, and image datasets can enhance multimodal applications (Liu et al., 2024).

- 2) Pretrained Model APIs: External AI services, such as GPT APIs or Google Bard, extend the LLM's capabilities by integrating specialized generative or analytical models (Xu et al., 2024).

The integration of RAG with LLMs introduces several notable advantages:

1. Factual Accuracy

RAG significantly reduces the risk of hallucinations by grounding responses in retrieved external data. For instance, a query about regulatory updates can reference the latest policy documents, ensuring factual correctness (Lewis, et al., 2020; Xu et al., 2024)).

2. Dynamic Adaptability

Unlike traditional LLMs, RAG systems do not require retraining to incorporate new information. By relying on external retrieval, they remain adaptable to evolving knowledge domains and datasets (Liu et al., 2024).

3. Enhanced Contextual Reasoning

RAG expands the effective context available to LLMs by integrating relevant external knowledge, enabling them to address complex and multi-faceted queries more effectively (Xu et al., 2024).

4. Scalability Across Domains:

The retrieval component can be tailored to domain-specific repositories, making RAG-equipped LLMs versatile for specialized applications (Lewis et al., 2020).

LLMs have revolutionized the field of natural language processing, offering unparalleled capabilities in semantic understanding, contextual awareness, and reasoning. Their ability to adapt to new tasks with minimal data and integrate with external knowledge sources makes them highly versatile tools for domain-specific applications. However, challenges like hallucination, lack of transparency, and the cost of adaptation underscore the need for complementary solutions, such as integrating LLMs with structured systems like knowledge graphs. Addressing these limitations will ensure that LLMs can deliver reliable, transparent, and efficient results in specialized fields like regulatory compliance and classification.

2.7 Integration of Knowledge Graphs with LLMs

2.7.1 Roadmap of Unifying LLM and Knowledge Graph

The objective of this review is to explore and evaluate the methods and frameworks that have been developed for integrating Large Language Models (LLMs) with Knowledge Graphs (KGs). The paper by Pan, et al., (2023) provides a comprehensive roadmap for unifying Large Language Models (LLMs) and Knowledge Graphs (KGs), addressing the limitations and strengths of both technologies. Outlined through three key frameworks shown as Figure 2.14:

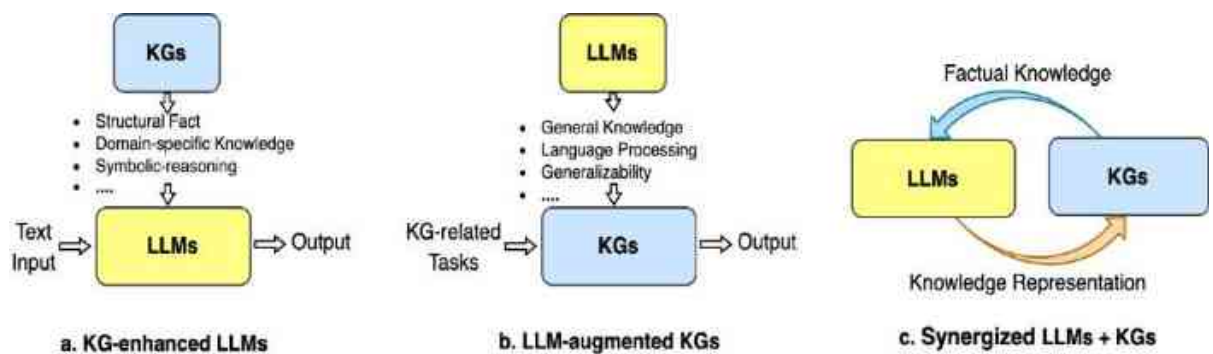


Figure 2.14 Outlined through three key frameworks shown road map of unifying LLM and Knowledge Graph proposed (Pan et al.,2023)

1. KG-Enhanced LLMs

This framework focuses on integrating KGs into LLM workflows to mitigate their limitations, such as hallucination, lack of domain-specific knowledge, and poor interpretability. By incorporating the structured factual knowledge stored in KGs, LLMs can achieve more accurate and contextually relevant outputs. The integration can occur during different phases:

- 1) **Pre-training:** KG information is embedded into LLM training objectives to enhance their understanding of relationships and factual data. KG information, such as hierarchical relationships within BTKI (e.g., Sections, Chapters, Headings, Subheadings), can be embedded into LLMs during pre-training. This

improves the LLM's ability to understand complex relationships in product classifications.

- 2) **Inference:** Techniques like Retrieval-Augmented Generation (RAG) retrieve relevant knowledge from KGs at runtime, enriching the LLM's output with up-to-date information. LLM queries the KG in real-time to retrieve relevant data about potential HS codes. For example, when classifying a "steel-reinforced hydraulic hose," the KG provides contextual insights about material and application hierarchies, refining the LLM's prediction.
- 3) **Interpretability:** KGs are used to probe and analyze LLM outputs, offering a transparent reasoning process that enhances trustworthiness, particularly in high-stakes scenarios. By grounding decisions in KG-derived facts, the LLM provides transparent justifications for its classifications. For instance, if an HS code prediction is questioned, the KG offers a traceable reasoning path (e.g., "classified as subheading 4009.12 based on material composition and function").
- 4) This framework addresses critical challenges like improving the factual accuracy of LLM outputs, grounding their reasoning in structured knowledge, and making them more interpretable.

2. LLM-Augmented KGs

In this framework, LLMs are employed to enhance the construction, representation, and utility of KGs. Traditional KGs, while accurate and structured, often suffer from issues like incompleteness and difficulty in representing unstructured textual knowledge. LLMs can address these gaps by:

- 1) **Embedding:** Enriching KG representations with semantic information derived from LLM processing of textual descriptions. For example, parsing terms like "hydraulic pump for excavators" to associate it with specific subcategories.
- 2) **Completion:** Inferring missing facts or relationships within the KG by leveraging LLM reasoning capabilities. LLMs infer missing relationships or attributes in the KG, such as linking new product descriptions to existing subheadings or updating outdated classifications in BTKI.

- 3) **Construction:** Automating the creation of KGs through entity discovery, relation extraction, and knowledge generation from unstructured data. LLMs extract entities and relationships from trade documents, customs records, or unstructured datasets to build or expand KGs. For instance, creating new nodes and edges for recently introduced BTKI regulations.
- 4) **KG-to-Text Generation:** Transforming structured KG information into natural language for better accessibility and usability. For example, generating readable justifications for why a "hydraulic pump" falls under a specific HS code.
- 5) **Question Answering:** Enabling LLMs to interact with KGs, bridging natural language queries and structured data reasoning. LLMs enable natural language querying of KGs, allowing users to ask complex questions like, "What subheadings apply to a hydraulic pump made of steel?"

This framework ensures that KGs remain dynamic, comprehensive, and capable of supporting complex reasoning tasks in real-world applications.

3. Synergized LLMs + KGs

The synergized framework envisions a bidirectional integration where LLMs and KGs mutually enhance each other. Unlike the previous frameworks that prioritize one over the other, this approach creates a unified system that combines the generative capabilities of LLMs with the structured reasoning strengths of KGs. Key features include:

- 1) **Bidirectional Reasoning:** Allowing LLMs to leverage KG knowledge for generative tasks while enabling KGs to benefit from LLM-driven insights for dynamic updates and reasoning. LLMs use KG knowledge for generative tasks like predicting HS codes, while KGs dynamically update using LLM-derived insights. For example, a product misclassified by the LLM can prompt KG updates to improve future predictions.
- 2) **Unified Data Processing:** Integrating both textual and structured data seamlessly for enhanced knowledge representation and reasoning. Combines structured data from KGs (e.g., BTKI hierarchies) with unstructured data processed by LLMs (e.g., product descriptions, trade documents). This ensures a seamless flow of information for accurate classification.

- 3) **Scalability and Flexibility:** The framework is designed to handle diverse applications, including hierarchical classification, search engines, recommendation systems, and beyond. The system can adapt to changes in BTKI regulations or evolving trade data, ensuring compliance and relevance.

2.7.2 Synergized KG and LLM Method in HS Code Classification

Large Language Models (LLMs) and Knowledge Graphs (KGs) represent two pivotal advancements in data science, particularly for semantic reasoning and hierarchical information retrieval. LLMs have proven their capability in understanding and generating natural language, while KGs offer structured and interpretable data representations. Integrating these technologies has opened new possibilities in automating complex classification tasks like Harmonized System (HS) code determination, a domain traditionally reliant on expert judgment and manual processes.

In the context of HS code classification, where part number (PN) structures and regulatory data (e.g., BTKI regulations) must be interpreted and matched, the synergy between LLMs and KGs becomes indispensable. By querying KGs using LLMs, users can retrieve regulatory and PN-specific information to assist in making accurate and context-aware judgments. This combination not only facilitates automation but also enhances transparency and explainability, critical for user trust in high-stakes applications. This section critically analyzes current methods integrating LLMs and KGs, focusing on their strengths, limitations, and applicability to HS code classification. By exploring existing frameworks like:

1. **Retrieval-Augmented Generation (RAG)**

RAG frameworks integrate an LLM with an external knowledge source to generate contextually accurate and factually grounded responses. This approach leverages the LLM's natural language understanding capabilities while using the knowledge source for grounding, reducing risks of hallucination and enhancing reliability. RAG can query a KG storing hierarchical regulatory information and PN details. For example, in an HS classification context, it could retrieve a product's specific BTKI regulations and PN mappings, providing both a suggested code and reasoning. Strengths: It builds LLM-driven knowledge graphs

to enable multi-hop reasoning and contextual retrieval. For instance, regulatory entities (e.g., tariffs, product-specific clauses) can be connected to PN-related nodes, allowing complex queries such as, “What HS code applies to spare parts under this regulation?” (Larson and Truitt, 2024).

1) Strengths of RAG in General:

- Efficient handling of user-specific queries.
- Enhanced contextual understanding, particularly when rules and hierarchies are explicitly modeled in KGs.
- Reduced hallucination risks by grounding LLM outputs in factual databases.

2) Limitations in Current Systems:

- Dependency on the quality and completeness of the underlying KG.
- Scalability issues when applied to large, complex datasets like HS code regulations spanning multiple industries.

2. Real-Time Query Generation

Real-time query generation systems use LLMs to transform natural language queries into structured database queries (e.g., Cypher for Neo4j). This approach facilitates direct interaction with graph databases without requiring users to have technical query-writing expertise. Applications in HS Code Classification: Incorporating such systems allows experts to query KGs with questions like: “What are the tariff rules for this PN under current BTKI regulations?” or “List all subcategories related to HS code 8409 for Indonesia.” Systems like those described by Hornsteiner et al., (2024) translate user queries into Cypher commands for databases. They also provide error correction and interactive feedback mechanisms for refining the query.

- 1) Strengths: Simplifies access to complex KGs, making them usable for non-technical practitioners. Provides flexibility in querying dynamically changing datasets.
- 2) Limitations: Query precision depends heavily on the LLM’s training and fine-tuning quality. Domain-specific challenges, such as ambiguous PN descriptions or overlapping regulations, require manual refinement of KG structures.

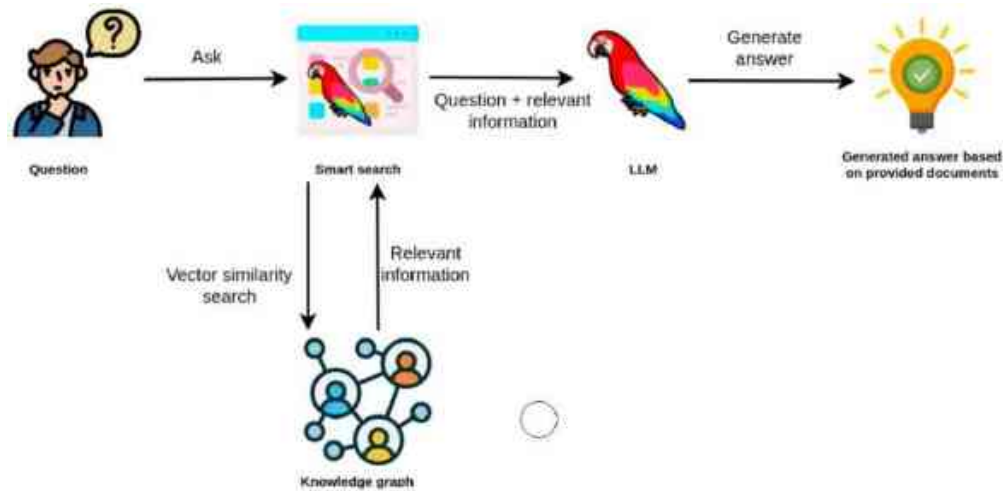


Figure 2.15 Retrieval-Augmented Generation Application to retrieve additional information from various sources such as Knowledge Graph (medium.com)

Integrating Knowledge Graphs (KGs) and Large Language Models (LLMs) offers significant potential for automating HS code classification shown Figure 2.15. KGs provide structured, interpretable knowledge critical for hierarchical classifications, while LLMs bring generative and reasoning capabilities. Current frameworks demonstrate promise but face challenges in scalability, domain-specific accuracy, transparency, and maintaining up-to-date knowledge. A unified, bidirectional integration of KGs and LLMs can address these challenges by leveraging KG structure for grounded reasoning and LLMs for dynamic updates. Research Gaps:

1. **Domain-Specific Integration:** Limited focus on complex domains like HS code classification for heavy equipment parts, where nuanced regulatory relationships are essential.
2. **Scalability:** Challenges in handling large, dynamic datasets such as BTKI regulations and evolving product catalogues.
3. **Transparency:** Insufficient mechanisms for traceable and interpretable decision-making to ensure regulatory compliance and user trust.
4. **Knowledge Representation:** Incomplete or outdated KGs and LLMs failing to reflect the latest regulatory updates and product information.
5. **Unified Bidirectional Systems:** Lack of frameworks fully leveraging the mutual strengths of KGs and LLMs for reasoning, updates, and scalability.

This framework aims to address these gaps through hierarchical knowledge embedding, LLM Retrieval-Augmented Generation (RAG), and dynamic bidirectional integration for accurate, transparent, and scalable HS code classification.

2.7.3 LangChain: Integrated Framework of LLM and Knowledge Graph for HS Code Classification

LangChain is a robust framework designed to facilitate the development of applications using Large Language Models (LLMs). When integrated with Knowledge Graphs (KGs), LangChain enhances the reasoning capabilities of LLMs by grounding them in structured data, thus addressing critical challenges in HS code classification. This synergy offers a scalable, adaptable, and efficient framework for automating complex tasks like HS code determination.

The integration of LLMs and KGs through LangChain enables a seamless workflow for natural language processing and structured data retrieval. The key components of this integrated framework are:

1. **Natural Language Query Processing:** LangChain converts user input into actionable queries through its prompt management and dynamic query generation modules. For example, a user query such as "What is the HS code for hydraulic pumps under BTKI regulations?" is transformed into a structured query compatible with the KG (Kiran et al., 2024; Korat, 2024).
2. **Knowledge Retrieval via Knowledge Graphs:** KGs store hierarchical and domain-specific data, such as BTKI regulations and product classifications. LangChain interacts with KGs using custom chains that enable multi-hop reasoning and context-aware data retrieval, ensuring accurate and relevant information is provided (Kiran et al., 2024; Korat, 2024).
3. **LLM-Driven Reasoning and Generation:** Once the KG retrieves data, the LLM processes the structured information to provide natural language explanations and classifications. This combination allows the system to generate HS code

suggestions along with transparent justifications, enhancing user trust (Kiran et al., 2024).

4. **Feedback and Iterative Refinement:** Users can interact with the system to refine queries or validate results. LangChain supports this iterative process by maintaining conversational context and integrating user feedback into subsequent queries (Korat, 2024).

Key Features of the Framework:

1. **Dynamic Query and Response Management:** LangChain bridges natural language input with KG-based structured data, dynamically fetching product-specific and regulatory information for HS code classification (Kiran et al., 2024; Korat, 2024).
2. **Multi-Hop Reasoning:** The combination of LangChain and KG enables complex queries, such as cross-referencing product categories with specific regulatory requirements, enhancing the system's reasoning capabilities (Kiran et al., 2024).
3. **Transparency and Explainability:** By leveraging KG structures, the framework provides traceable and interpretable outputs, critical for compliance and decision-making in regulatory environments (Korat, 2024).
4. **Few-Shot Learning and Customization:** LangChain supports few-shot learning and prompt engineering, allowing the system to adapt to domain-specific nuances, such as variations in product descriptions or overlapping regulations (Korat, 2024).

Applications in HS Code Classification:

1. **Hierarchical Classification:** The framework facilitates hierarchical reasoning, mapping part numbers and product descriptions to specific HS codes within the BTKI system.
2. **Regulatory Compliance:** Automated checks ensure alignment with up-to-date regulatory standards by querying KGs for changes in tariffs and classifications.

3. **User Interaction and Refinement:** Non-technical users can interact with the system using natural language, refining queries for complex cases like multi-component heavy equipment.

This integrated framework addresses key challenges in HS code classification:

1. **Scalability:** Efficient handling of extensive datasets and dynamic updates, such as evolving BTKI regulations and expanding product catalogs.
2. **Transparency:** Traceable workflows through KG integration and LangChain's modular architecture.
3. **Domain-Specific Adaptability:** Customizable workflows tailored to industries like heavy equipment parts, where nuanced regulatory relationships are essential.

2.7.4 Evaluation Framework for KG-LLM Systems

The Retrieval Augmented Generation Assessment (RAGAS) framework is a reference-free evaluation method designed to assess Retrieval-Augmented Generation (RAG) pipelines comprehensively. These systems, combining a retrieval mechanism and a generation module, aim to leverage external databases to enhance the output of large language models (LLMs). RAGAS addresses key evaluation dimensions critical to the RAG architecture: retrieval accuracy, generation fidelity, answer relevance, and context recall (Shahul et al., 2023).

Key Evaluation Dimensions:

1. **Faithfulness**

Faithfulness measures whether the generated answers are accurately grounded in the retrieved context. The process involves breaking down responses into distinct statements and assessing their alignment with the source context. The framework employs LLMs to validate each assertion's fidelity using predefined prompts, culminating in a faithfulness score calculated as the ratio of supported statements to the total (Shahul et al., 2023).

2. **Answer Relevance**

This metric evaluates how well the generated responses align with the original queries. RAGAS uses LLMs to create alternative queries based on the output and computes their semantic similarity to the initial question using vector-based embeddings. The score reflects the average similarity, ensuring relevance to user-provided inputs (Shahul et al., 2023).

3. Context Relevance

Context relevance determines whether the retrieved passages are focused and directly support the task at hand. The evaluation penalizes inclusion of extraneous data, optimizing resource efficiency and retrieval precision. The system assesses this dimension by analyzing the proportion of meaningful information relative to the overall content retrieved (Shahul et al., 2023).

4. Context Recall

Context recall measures the extent to which relevant passages from the database are retrieved. It evaluates whether the retrieval mechanism has successfully retrieved all essential information necessary for the generation module. This metric is critical for applications requiring complete context coverage, as incomplete retrieval can compromise the generation's accuracy. RAGAS computes this by comparing the overlap between retrieved passages and a theoretical set of relevant passages identified using benchmarks or heuristics (Shahul et al., 2023).

The RAGAS framework supports the evaluation of pipelines without requiring pre-existing ground truth annotations. It integrates with widely adopted frameworks like Langchain and llama-index, ensuring adaptability in various RAG implementations. Using automated and systematic evaluation prompts, the framework enables efficient, iterative testing of RAG architectures under multiple constraints (Shahul et al., 2023).

2.7.5 Large Language Models in Knowledge Graph Information Retrieval

The integration of Large Language Models (LLMs) with Knowledge Graphs (KGs) has emerged as a powerful approach to enhancing the capabilities of artificial intelligence systems. This essay critically compares previous studies, examining their

methodologies, applications, and relevance to domains such as HS Code classification as seen in table 2.10:

Table 2.10 Related Works Integration LLM and Knowledge Graph

Paper	Objective	Methodology	Tools/ Technologies	Research Gap
Zhao et al. (2024) - AGENTiGraph	To integrate LLMs with KGs for real-time, user-friendly interaction and task management.	Multi-agent architecture for task classification, semantic parsing, and knowledge integration.	AGENTiGraph, Neo4j, Few-Shot Learning, Chain-of-Thought reasoning.	Limited domain-specific applications and scalability for complex queries.
Chen et al., (2021) - Prometheus Chatbot	To enhance recommendation systems using LLMs and hybrid KG construction.	Combines top-down and bottom-up KG construction with GPT-4 for generating recommendations.	Neo4j, Azure OpenAI, LangChain.	Generalizability to domains beyond computer components.
Hornsteiner, et al, (2024)	To automate Cypher query generation from natural language inputs.	Modular pipeline with error-checking mechanisms for query validation.	Neo4j, GPT-4 Turbo, Cypher Query Language.	Limited adaptation to unstructured data and diverse domain-specific schemas.
Pusch and Conrad, (2024)	To reduce hallucinations in LLM-based systems by validating outputs against KG schemas.	Query-checking algorithm with retrieval-augmented generation (RAG) and schema-based validation.	LangChain, GPT-4 Turbo, llama3:70b, Cypher queries.	High computational cost and need for optimization in non-biomedical contexts.
Abu-rasheed, et al., (2024)	To provide explainability in learning recommendations using KG-enriched LLM responses.	Multi-step conversational explainability via contextualized prompts.	GPT-4, KG-based contextualization, mentoring chatbots.	Limited exploration of task scalability and real-time performance.

Paper	Objective	Methodology	Tools/ Technologies	Research Gap
Dam, et al., (2024)	To survey LLM-based chatbots across industries and identify challenges and applications.	Literature review and taxonomy of challenges and applications.	Literature analysis, comparative framework.	Lack of actionable frameworks for domain-specific implement

The integration of Knowledge Graphs (KGs) with Large Language Models (LLMs) offers promising solutions for challenges in HS Code classification, particularly in improving interaction, reliability, and explainability. This section synthesizes key studies to outline potential enhancements.

1. Enhancing Query Interaction and Flexibility

(Zhao *et al.*, 2024) introduced the AGENTiGraph framework, enabling real-time, multi-agent interaction with KGs to resolve factual inconsistencies and improve LLM reasoning. Its modular adaptability and visualization capabilities make it suitable for dynamic domains like regulatory compliance. Similarly, (Hornsteiner et al. (2024)) proposed a Real-Time Text-to-Cypher Query Generation system, translating natural language queries into Cypher commands with error-checking modules for precision. While AGENTiGraph focuses on adaptability, Hornsteiner's approach emphasizes structured query accuracy. For HS Code classification, integrating AGENTiGraph's modularity with Cypher generation's precision could enhance the ability to handle diverse customs data.

2. Reducing Hallucinations and Ensuring Reliability

Pusch and Conrad (2024) addressed LLM hallucinations by validating Cypher queries against KG schemas, ensuring syntactic and semantic alignment. This approach, proven effective in high-stakes domains like biomedicine, offers a robust mechanism to mitigate errors. Conversely, Chen et al. (2021) developed the Prometheus Chatbot, which integrates hybrid KG construction to enhance data representation for computer component recommendations. While Pusch and Conrad's schema validation ensures reliability, the hybrid approach of

Prometheus could address HS Code classification's need to harmonize textual product descriptions with regulatory data.

3. Promoting Explainability and User Trust

Explainability is pivotal for stakeholder trust in HS Code recommendations. Aburashed et al. (2024) leveraged KG-enriched prompts in their Learning Recommendation Chatbot to contextualize responses and foster transparency. Similarly, AGENTiGraph's semantic parsing simplifies complex queries into actionable graph tasks, enhancing user comprehension (Zhao *et al.*, 2024). Combining these approaches could provide explainable recommendations for HS Codes, ensuring stakeholders understand the rationale behind each classification.

4. Addressing Domain-Specific Challenges

Domain-specific customization remains a significant challenge. The Prometheus Chatbot employs a hybrid KG approach combining top-down modeling with bottom-up data extraction, making it ideal for constructing domain-specific knowledge bases like those required for HS Code classification (Chen et al. (2024), 2021) However, broader challenges in scalability and domain adaptation highlighted by (Dam et al., 2024) need to be addressed to optimize these systems for real-world application.

2.8 Risks and Challenges of AI in HS Code Classification

The integration of Artificial Intelligence (AI) into HS code classification systems has the potential to transform efficiency and accuracy in global trade. However, this innovation is not without challenges. From data quality and systemic risks to operational and ethical considerations, each risk area presents unique obstacles that must be addressed to ensure the successful deployment of AI-driven solutions. These risks are particularly relevant within the nuanced regulatory context of Indonesia's BTKI framework, where precision and compliance are paramount.

2.8.1 Data-Related Risks

AI systems rely heavily on data for their effectiveness, and the quality and representativeness of that data play a critical role in shaping the outcomes and risk:

1. **Data Quality Issues:** AI systems depend on high-quality datasets for effective model training and predictions. Challenges such as incomplete product descriptions, outdated regulatory data, and inconsistencies across datasets can lead to misclassifications. These errors have significant consequences, including customs clearance delays and non-compliance penalties (Lee et al., 2024).
2. **Bias in Training Data:** Bias in training data arises from overrepresentation or underrepresentation of certain product descriptions or regulatory contexts. This can result in systematic classification errors, particularly when aligning global trade practices with local regulations. Addressing this risk requires diverse and representative datasets that reflect the variations in interpretations and rules across regions (P Slattery, 2024).

2.8.2 Systemic Risks

The complexity of AI models introduces several systemic risks that can hinder their application in dynamic and regulated environments as follows: ®

1. **Black-Box Models:** AI models, particularly those based on deep learning, often operate as "black boxes," lacking transparency in their decision-making processes. This lack of interpretability is a major challenge in regulatory contexts where decision explanations are crucial for compliance. Techniques like SHAP and LIME can help provide insights into model decisions, making them more interpretable and aligned with hierarchical classification systems (Lee et al., 2024).
2. **Overfitting:** occurs when models become too tailored to be training data, reducing their ability to generalize to new datasets. This is a critical risk in HS Code classification, where evolving regulations or the introduction of new product categories may render existing models ineffective. Continuous monitoring and retraining are necessary to mitigate this issue (Martino et al., 2022).

2.8.3 Operational Risks

The practical deployment of AI in customs and trade introduces operational risks that can disrupt workflows and compliance efforts as follows:

1. **Compliance Challenges:** Misclassifications can lead to regulatory violations, penalties, and disruptions in supply chain operations. For instance, classifying products under incorrect HS codes may delay shipments, increase costs, and damage stakeholder trust. Ensuring compliance with BTKI regulations necessitates rigorous validation mechanisms (Lee et al., 2024).
2. **Latency and Efficiency:** Real-time customs clearance workflows demand low-latency decision-making from AI systems. High computational requirements for processing hierarchical classifications can create bottlenecks, impeding operational efficiency. Optimizing AI models for speed and reliability is essential to maintain seamless workflows (Martino et al., 2024).

2.8.4 Ethical Risks

The ethical implications of AI-driven classification systems must be carefully considered to ensure fairness and accountability:

1. **Bias in Outputs:** AI outputs may perpetuate biases inherent in training data, potentially leading to unfair classifications. This risk is particularly concerning when underrepresented regions or products are misclassified, disadvantaging certain stakeholders. Addressing bias requires ongoing auditing and fairness monitoring (P Slattery, 2024).
2. **Automation Risks:** While AI systems aim to streamline operations, they risk reducing reliance on human expertise. However, the goal should not be to replace experts but to enhance their capabilities. By complementing human judgment with AI-driven insights, organizations can maintain accountability while improving efficiency (Martino et al., 2024).

2.8.5 Role of Human-in-the-Loop (HITL)

Human-in-the-Loop (HITL) systems play a critical role in addressing the risks associated with AI in HS code classification. By incorporating human oversight, HITL workflows ensure that AI systems remain reliable, adaptable, and aligned with regulatory requirements. Following are some advantages of HITL to overcome some of the risk:

1. **Regulatory Complexity:** The hierarchical and nuanced nature of the BTKI framework often requires expert judgment for accurate classifications. AI systems may struggle to fully capture regulatory subtleties, making HITL workflows essential for validation and compliance (Lee et al., 2024).
2. **Ambiguity Resolution:** Low-confidence classifications often arise in cases where product descriptions are incomplete or ambiguous. HITL workflows enable domain experts to review these outputs and ensure their accuracy, reducing the likelihood of errors (Lee et al., 2024).
3. **Feedback Mechanism:** HITL workflows provide an iterative feedback loop that improves AI systems over time. Expert reviews of misclassifications can help refine model algorithms and update Knowledge Graphs, ensuring alignment with evolving regulatory and operational requirements (Lee et al., 2024).

The application of AI in HS code classification within Indonesia's BTKI framework is complicated by the reliance on tacit knowledge and subjective interpretation of regulations. These challenges emphasize the need for precise and adaptable systems:

1. **Tacit Knowledge Incorporation:** There is insufficient research on how AI systems can encode and operationalize tacit knowledge, particularly for nuanced BTKI classifications. Hybrid approaches that combine AI with human expertise (e.g., Human-in-the-Loop systems) need further exploration.
2. **Handling Ambiguity and Interpretation:** Current AI models lack mechanisms to address interpretation variability in BTKI classifications. Research is needed on leveraging knowledge graphs and context-aware models to manage ambiguous or incomplete data.
3. **Adapting to Regulatory Changes:** Few studies address dynamic AI frameworks capable of integrating real-time regulatory updates, crucial for compliance with evolving BTKI rules.
4. **Explainable AI (XAI) in BTKI:** There is limited research on explainability in the context of AI for HS code classification under BTKI. Regulatory compliance often requires interpretable decisions, yet current systems prioritize accuracy over transparency.

5. **Bias in Local Contexts:** Research on how bias in training datasets impacts AI performance for region-specific regulations like BTKI is scarce. Addressing this requires localized datasets and fairness-oriented algorithms.

The gaps highlight the need for context-aware AI systems that can complement human tacit knowledge, handle variability in regulatory interpretation, and adapt to Indonesia's evolving BTKI framework.

2.8.6 Risks in HS Code Classification at PT XYZ

In this table 2.11 summary and categorizes AI-related risks for PT XYZ, assesses their severity, and outlines mitigation strategies to enhance compliance, accuracy, and operational efficiency in AI-driven HS code classification. The analysis underscores the importance of a hybrid AI approach integrating Knowledge Graphs (KGs), Large Language Models (LLMs), Explainable AI (XAI), and Human-in-the-Loop (HITL) workflows.

Table 2.11 Risks in HS Code Classification at PT XYZ

Risk Category	Identified Risks	Challenges for PT XYZ	Risk Level	Potential Mitigation Strategies
Data-Related Risks	Data Quality Issues	Incomplete product descriptions and outdated regulatory data lead to misclassifications and customs clearance delays.	High	Improve data preprocessing, use Knowledge Graphs (KGs) for structured data enrichment.
	Bias in Training Data	Overrepresentation or underrepresentation of certain products or regulations causes systematic errors.	High	Use diverse, region-specific datasets; implement fairness checks in AI training.
Systemic Risks	Black-Box Models	Lack of interpretability in AI decisions makes compliance verification difficult.	High	Integrate Decision Support System with Human In the Loops
	Overfitting	AI models struggle to generalize to new products and evolving HS codes, leading to inaccuracies.	Medium	Implement continuous retraining with real-time trade data updates.

Risk Category	Identified Risks	Challenges for PT XYZ	Risk Level	Potential Mitigation Strategies
Operational Risks	Compliance Challenges	Misclassifications result in regulatory violations, penalties, and supply chain disruptions.	High	Human-in-the-Loop (HITL) validation to oversee AI recommendations.
	Latency and Efficiency	AI processing delays slow down real-time customs clearance.	Medium	Optimize AI models for low-latency decision-making and computational efficiency.
Ethical Risks	Bias in Outputs	Underrepresented product categories may be misclassified, disadvantaging specific industries.	High	Regular bias audits and fairness monitoring for AI predictions.
	Automation Risks	Over-reliance on AI could reduce human expertise and accountability.	Medium	Maintain a decision support system (DSS) that balances automation with expert oversight.
Regulatory Adaptation Risks	Regulatory Complexity	AI struggles to interpret evolving BTKI regulations, leading to non-compliance.	High	Use Knowledge Graphs for real-time rule updates and compliance tracking.
	Tacit Knowledge Incorporation	AI lacks the ability to encode domain-specific knowledge held by experienced customs officers.	High	Implement Human-in-the-Loop (HITL) workflows for expert validation.
	Handling Ambiguity and Interpretation	AI models fail to distinguish between closely related HS codes when descriptions are vague.	High	Combine Large Language Models (LLMs) with KGs for semantic understanding.
	Explainability and Traceability	AI-generated classifications lack traceability, making audits and dispute resolution difficult.	High	Implement Explainable AI (XAI) to provide justifications for AI decisions.

2.9 Proposed Framework

2.9.1 Synthesis of Findings

The limitations of machine learning (ML) in HS code classification including its inability to handle hierarchical structures, regulatory evolution, contextual ambiguity,

and lack of explainability—highlight the need for more advanced AI-driven approaches. While ML improves efficiency, it alone cannot ensure accuracy, traceability, and adaptability in a complex, high-stakes trade environment. To address these gaps, Large Language Models (LLMs) and Knowledge Graphs (KGs) present promising solutions, enhancing contextual understanding, hierarchical classification, and regulatory compliance in a robust decision support system (DSS).

1. Addressing Data Quality and Bias with Knowledge Graphs

One of the primary challenges in AI-driven classification is inconsistent and incomplete product descriptions, leading to misclassification and regulatory non-compliance. Traditional ML models struggle to reason beyond structured data, making them ill-suited for handling ambiguous classifications. KGs provide a structured, rule-based approach, helping to:

- Encode hierarchical relationships within HS codes, ensuring consistent classification decisions.
- Reduce data bias by integrating diverse trade data sources, improving representation across regulatory frameworks.
- Support explainability, allowing AI models to justify classifications through explicit knowledge structures and traceable decision pathways.

By incorporating KGs as a foundational layer, AI models can achieve greater interpretability and accuracy, particularly when handling region-specific trade rules such as Indonesia's BTKI framework.

2. Enhancing Context Awareness with Large Language Models (LLMs)

HS code classification requires deep contextual understanding that traditional AI models lack. LLMs, trained on multilingual trade documents and regulatory texts, offer a solution by:

- Interpreting ambiguous product descriptions, reducing misclassifications caused by vague or incomplete data.
- Identifying semantic relationships between product descriptions and HS codes, improving context-aware classification.

- Providing adaptive learning, allowing AI systems to process regulatory updates in real-time.

However, LLMs alone are insufficient due to their lack of domain-specific constraints. Combining LLMs with KGs ensures that AI-driven classifications remain structured, explainable, and aligned with evolving regulations.

3. Ensuring Transparency and Compliance with Human-in-the-Loop Systems

Regulatory compliance in HS code classification demands transparency, as black-box AI models are unsuitable for legally binding decisions. The Human-in-the-Loop (HITL) approach ensures AI remains aligned with human expertise and regulatory oversight, addressing challenges such as:

- **Regulatory complexity:** AI can recommend classifications, but human reviewers validate high-risk or low-confidence cases, ensuring compliance with BTKI regulations.
- **Ambiguity resolution:** HITL workflows allow customs officers to refine AI-generated outputs, reducing classification errors in borderline cases.
- **Feedback loops:** Expert corrections are fed back into AI models, improving classification accuracy over time while ensuring alignment with trade policies.

4. Managing AI Risks Through Explainable AI (XAI) and Traceability

Regulatory frameworks such as BTKI require AI models to be transparent and interpretable. Explainable AI (XAI) such as decision support system with LLM and KG, help:

- Provide justifications for AI-generated classifications, **allowing** auditors to verify decisions.
- Reduce user distrust, ensuring that AI-driven processes remain accountable and reliable.
- Improve model robustness, allowing AI systems to adapt to trade policy changes while maintaining traceability of past decisions.

Given the complex regulatory landscape, hierarchical HS code structures, and need for human oversight, a fully automated AI system is not feasible. Instead, a hybrid AI framework integrating Knowledge Graphs, Large Language Models, and Human-in-the-Loop systems provides the most viable approach. This framework would:

- Enhance classification accuracy by leveraging context-aware AI models.
- Ensure regulatory compliance through explainability and traceability mechanisms.
- Adapt to evolving trade regulations, reducing the need for frequent manual intervention.

By adopting this approach, PT XYZ can mitigate AI risks while maximizing the efficiency, transparency, and compliance of HS code classification under Indonesia's BTKI framework.

2.9.2 Relevance of Proposed Framework

The synthesis of findings underscores the need for a unified framework that integrates the complementary strengths of ML, KGs, and LLMs to address the specific challenges of HS code classification under the BTKI framework. The proposed framework integrates three primary components to meet the specific demands of the heavy equipment industry:

1. **Machine Learning for Automation:**

ML models automate the classification of large volumes of structured product descriptions, such as part numbers (PNs) and standardized specifications, common in the heavy equipment sector. These models quickly identify preliminary HS codes for parts like hydraulic pumps, engine filters, and excavator components, reducing the need for manual processing.

2. **Knowledge Graph + LLM for Retrieval and Hierarchical Reasoning:**

Knowledge graphs encode the hierarchical relationships and regulatory rules of the BTKI, ensuring accurate classification of heavy equipment parts within the appropriate sections, chapters, headings, and subheadings. LLMs enhance reasoning by interpreting free-form product descriptions, resolving ambiguities,

and querying the KG for contextual and regulatory insights. For instance, determining whether a hydraulic pump should be classified under "machinery parts" or "hydraulic systems" based on material composition and intended use.

3. LLM for Transparency and Expert Interaction:

LLMs provide detailed explanations for each classification, helping customs officials and industry experts validate or refine classifications. This ensures that stakeholders can confidently navigate complex regulations, such as distinguishing between subcategories for steel versus aluminium components.

2.9.3 Framework Workflow

The proposed workflow consists of five interconnected steps:

1. **Input:** The process begins with raw product descriptions, which may include structured data (e.g., part numbers, specifications) and unstructured text (e.g., narrative descriptions).
2. **ML Classification:** Machine learning models analyze the input data and assign preliminary HS codes based on patterns learned from training datasets. This layer handles broad-level classifications efficiently, reducing the volume of cases requiring detailed review.
3. **KG Query via LLM:** For ambiguous or hierarchical classifications, the LLM queries the KG to retrieve relevant information. For instance, the KG may provide insights into Part number regulatory notes, material composition rules, or functional categorizations that refine the ML-assigned HS code.
4. **Expert Interaction:**
5. The LLM generates a detailed explanation of the classification process, including the reasoning derived from the KG and the ML model. Experts can validate, override, or refine the classification based on this information.
6. **Final Output:** The final step outputs an accurate HS code assignment, along with a transparent explanation and any relevant regulatory notes. This ensures compliance and provides traceability for audit purposes.

2.9.4 Validation Framework through Process Modelling Success (PMS)

The Process Modelling Success (PMS) Framework, introduced by Sedera et al. (2002), provides a structured approach to evaluate the effectiveness of process modelling initiatives. It focuses on key stages, from design to organizational impact, and aligns perspectives from modellers and end-users. This framework is ideal for validating the proposed integration of Machine Learning (ML), Large Language Models (LLMs), and Knowledge Graphs (KGs) in automating HS code classification. The validation will be conducted through Focus Group Discussions (FGDs) with domain experts.

The PMS Framework in figure 2.16 measures success across 5 dimensions. Each dimension offers unique insights into how the system performs during design, implementation, and real-world application:

1. **Modeler Satisfaction:** Focuses on the experience, it evaluates user satisfaction with resources, tools, and processes used, particularly LLM integration with domain-specific KGs.
2. **Process Model Quality:** Assesses the outputs of the system, including the accuracy of HS code classification, semantic and syntactic correctness, and clarity of results. This dimension is critical to ensure that the system aligns with the BTKI regulatory framework and provides actionable insights.
3. **Model Use:** Examines the system's real-world application by end-users, such as customs officers and compliance managers. It evaluates usability, adaptability to regulatory changes, and the system's ability to reduce manual efforts in HS code classification.
4. **Process Impact:** Evaluates the broader effects on workflows and operational efficiency. This includes the system's ability to improve compliance, reduce classification errors, and support organizational goals like cost reduction and enhanced transparency.
5. **Organizational Impact:** Examines the broader benefits for the organization, such as improved operational efficiency and strategic decision-making.

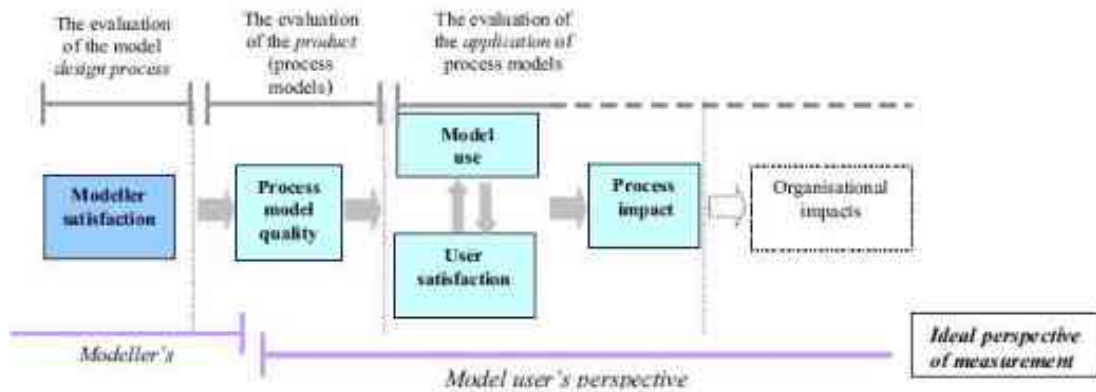


Figure 2.16 Process Modelling Success (PMS) measurement framework (Sedera Rosemann and Gabe,2002)

The PMS Framework will guide the validation process during FGDs. The focus will be on demonstrating the proposed system to experts, gathering feedback, and refining the solution based on insights. The discussions will address the following:

1. **System Demonstration:** A walkthrough of the system's key features, including its ability to classify HS codes using ML, LLMs, and KGs, and to handle complex queries such as mapping BTKI regulations to product details.
2. **Expert Evaluation:** Experts will assess the system based on the four dimensions of the PMS Framework. Their feedback will highlight strengths, such as efficiency and usability, as well as areas for improvement.
3. **Continuous Refinement:** Insights from the FGD will be used to enhance the system's functionality. Particular focus will be given to improving semantic correctness, user interface, and compliance with evolving regulations.

This proposed framework provides a comprehensive solution for HS code classification, addressing scalability, transparency, and compliance challenges. The subsequent chapter (Chapter 3) details the methodology for implementing this framework, including data collection, model selection, KG construction, LLM and KG integration and evaluation metrics. By operationalizing the framework, the methodology chapter will demonstrate its practical application and effectiveness in real-world scenarios.