National College of Ireland

HS Code classification using Agentic AI for goods import and export

MSc Research Project
Data Analytics

Pathmanathan Gawtham
Student ID: 23386410

School of Computing
National College of Ireland

Supervisor: Athanasios Staikopoulos

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Pathmanathan Gawtham
Student ID:	23386410
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Athanasios Staikopoulos
Submission Due Date:	11/12/2025
Project Title:	HS Code classification using Agentic AI for goods import and export
Word Count:	11233
Page Count:	27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Pathmanathan Gawtham
Date:	25th January 2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

HS Code classification using Agentic AI for goods import and export

Pathmanathan Gawtham
23386410

Abstract

Currently within the Import/Export domain the tax code assignment of the products is handled either manually or in a semi-automated process which is time consuming and heavily dependent on domain experts. The products are also sent with improper or vague description that can cause delays in import/export. This paper explores how effectively the application of Agentic AI will classify the Harmonized System Code for import/Export of goods around the globe based on the description of the products which can improve the overall import/export process. The paper also provides a benchmark comparison by evaluating accuracy, precision, f1, recall and latency with an alternative methodology of Retrieval Augment Generation (RAG) combined with LLM. The main model that was used for implementing the Agentic AI is "gpt-04-mini". The agentic implementation was implemented and tested with three different datasets that are similar in nature. The research provides valuable insight into the limitations and challenges of using Agentic AI for compliance and tax code classification. The proposed methodology was able attain approximately 90% accuracy on unknown HS Code descriptions and 73% accuracy for synthetic dataset with sales like product descriptions. Based on the outcomes of the research the research questions were addressed and showcased potential application of Agentic AI were identified for future research. The paper concludes by strongly suggesting that use of AI can enforce Tax Compliance efficiently.

1 Introduction

In recent years there have been significant impact on import/export process due to changes in tariff rates applied between countries. This has put more pressure on custom official to comply with regulations and assign the correct set of tax codes and tariff rates. However, most of the import/export process is either done manually or semi-automatically with the help of experts which results in delays, misclassification of tax codes and regulatory issues(Chen; 2025). Utilizing the AI technology, we can make significant advancement in addressing issues and limitation in tax compliance(Bajpai; 2024). Many time-consuming tasks such as generating tax report, calculating tax liabilities and classifying transactions can be automated through an AI-driven tax system (Dudu et al.; 2024).Up until this point, traditional machine learning, Deep Learning and RAG applications have been used but they are not able to achieve a strong result to use in the real world due to the technical limitations.

For this research we identified a challenging aspect of Customs import/export process, which is assigning the product with the correct Harmonized System (HS) codes. These HS

Code are 6-digit codes with hierarchically structure representing a category for a relevant product(Amel et al.; 2023) . The first two digits represents Chapter of the HS code, if we combine the first 4 digit it represents the Heading of the HS code and finally the 6 Digits represents the actual HS code itself(Chen; 2025). The following diagram visualises how the HS code hierarchical structure is and what each hierarchical level represents within the HS Code

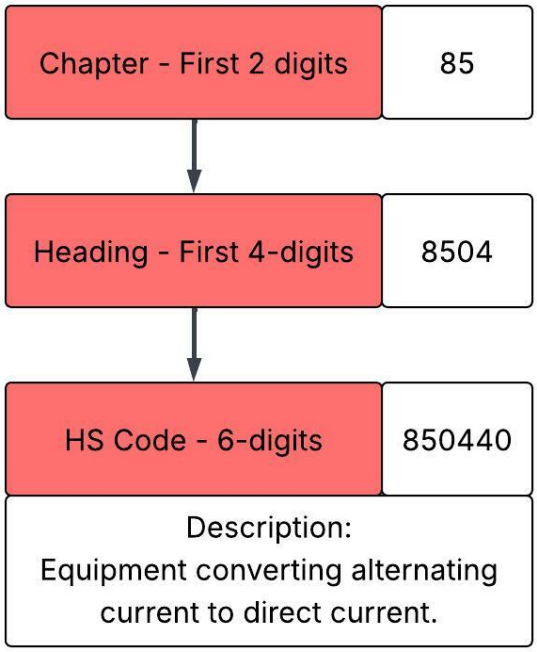

Figure 1: HS Code Hierarchical Structure

Many customs officials spend time interpreting the product descriptions and figuring out the correct HS codes based on the information given by their clients(Amel et al.; 2023). This is due to the formal structure of the text used in HS code description, which are distinct to the description of goods imported/ exported in real world (Liao et al.; 2024). Since this a text classification problem, many prior researchers have explored and applied different methodologies such as RAG, LLMs and Deep Learning models to assign the HS codes based on product descriptions. Despite producing promising results, they cannot move towards a real-world implementation due to many limitations and the primary limitations among them is, these models are depending on one dataset which they split to train, test and validate. They have not tested their model or methodology with different set of datasets to simulate real world variability and prove that these methodologies will work in real-world application. Even if they need to test with a new dataset, they will have to retrain the model again which is time and resource consuming. Hence, we wanted to explore a different methodology to tackle this classification problem and recently the potential showcased by Agentic AI framework motivated us to explore its use in classifying the HS codes based on the product description. Following are the research questions,

- RQ1 – How can a state-of-the-art LLM be utilized for Harmonized System code classification in import/export of goods?

- RQ2 – How does Agentic AI yields results when compared with RAG and RAG with LLM in terms of accuracy, precision, f1, recall?
- RQ3 – How effectively does Agentic AI performs when testing for generalization using an external dataset and synthetic dataset?
- RQ4 – How can the application of Agentic AI improve the tax compliance in import and export processes?

This study mainly focuses on RQ2, which experitments whether Agentic reasoning actually offers added value over retrieval-based methods when dealing with realistic, real-world language. RQ1, RQ3, and RQ4 play a supporting role by helping to frame the discussion around practicality, generalization, and compliance-related considerations.This research explores the potential and the limitations of using Agentic AI for tax code classification for import and export and contribute to the ongoing and future research on AI enforced tax compliance. Our research not only implement an Agentic AI workflow to classify the Harmonized system codes it also compares tree different approaches RAG, RAG and LLM and then the Agentic Flow where we evaluated and compared Accuracy, F1, Precision and Recall showing the performance difference between each approach. The research follows an empirical and experimental approach with combination of different parameters such as number of retrievals; prompt-engineering and different ways of vector embeddings and this research validates if the approach is generalized enough to perform well on unknown data.

The report consists of main sections and several subsections, following the Introduction section the Related Works section reviews preceding academic studies and critical analysis of their methodologies and findings. Then Research Methodology section outlines the step-by-step process that we performed for our research which can be used by future research to replicate and improve. Next section will be Design Specification which will contain the architecture and specification of our Agentic implementation followed by Implementation section which will contain the technical aspects of our research of the final stage and reasoning behind use of the tools. Then the Evaluation section will contain the different experiments conducted along with their results, followed by discussion from them. The final section Conclusion and Future Works will summarize our research findings, the limitations, potential improvements the future research and the conclusion to our research followed by the References section.

2 Related Work

Several prior research studies have been published related to applying AI technology in tax compliance and they have presented different perspective on different areas where tax compliance is required such as Fintech, Health Care, Customs Import/ Export, CRM, ERP systems, etc. The following sections will explore and provide a systematic literature review on various aspects of the domain related to our project.

2.1 Tax Compliance

In general, there have been many publications related to Tax Compliance proving and arguing the integration of latest technology to enhance the compliance process effectively

and integration of AI-driven compliance system has gained attention in recent years. A recent scholarly publication by Dr. Amardeep Bajpai covered the application of AI for financial regulation and tax compliance and one of their conclusions regarding tax compliance and AI is that different aspects of tax compliance such as detecting tax fraud and evasion, administration process and efficient allocation of resources for enforcement can be streamlined with integration AI(Bajpai; 2024). They have also provided evidence of successful AI integration in tax compliance that has yielded positive results from previous publications, which made their argument stronger and helped us with accessing our research.

Another similar publication by Rida Balahouaoui and El Houssain Attak conducted an extensive systematic review on the same topic using the Preferred Reporting Item for Systematic Reviews and Meta-Analyses methodology where they identified trends and insights by analyzing the textometry of titles, abstracts and keywords(Belahouaoui and Attak; 2024). Their study also revealed two main findings, digitalization of tax enhances the operation efficiency and tax compliance but adoption of AI and blockchain driven system faces challenges that require support from government body(Belahouaoui and Attak; 2024). From their systematic review we understood several potential areas where tax compliance is a challenging process and requires AI and block chain integration to increase efficiency.

Most recent publications have spoken in favor for an AI driven tax compliance framework or system, however there are few publications that has raised concerns. For example, research done by N. Kerinc LL.M and M. Serrat Romaní, they acknowledge the potential of the AI-driven system in taxation, but they also raise concerns when implementing such systems (LL.M. and Romaní; 2022). According to their study, the vast amount of tax data is biased and contains subjective terminologies that can lead to issues such as miss classification of tax codes or indirect discrimination when detecting fraud or tax evasion (Ezeife; 2021). They also emphasize being compliant with data regulatory bodies such as GDPR when using the data to train the AI models and validate the models to perform reliability under generalized conditions(Ezeife; 2021). One of the research questions of our study is to validate our implementation for generalization conditions and the dataset we use follows the GDPR compliance, which means our research findings provide stronger foundation and validation on applying our research for tax compliance. The previous research published related to tax compliance gave us an understanding of current state and the future it's headed for. By analyzing their methodologies and findings we established and re-evaluated our goals and ideas to be in line with their work to build a strong foundation for our work.

2.2 AI Applications in Customs

Many AI applications have been developed to enhance the Customs process such as Import/ Export, Tariff applications and automated commodity code assignments. Recently researchers from a prominent tech company, IBM, published their work on assigning HS codes based on description of the product. Their work is a direct foundation for our research and provides a baseline for improvement. They have proposed an ensemble-based approach by combining hybrid machine learning with knowledge-retrieval to assign the HS Codes (Shubham et al.; 2023). They have done thorough research related to HS Code hierarchical structure and created two models, where one of them is trained with just the 6 digits of HS code and the other trained with hierarchical context, when validated both

models scored accuracy of 56% and 65% respectively(Shubham et al.; 2023). The resulting accuracy is lower than the accuracy from our research, which is 70%, which proves our proposed method outperforms their method. Their method is limited due to the use of weak embeddings such as BERT-base, SBERT and rule-based retrieval with knowledge graphs. However, their use of knowledge graphs produces an excellent auditing method that can be used in real world, our research does not cover the auditing aspect of HS code and only focuses on improving assign HS codes, in future we can create a complete agentic flow with auditing.

A recent publication by Pamela Bayona states that they are working on HS code transposition using TF-IDF cosine similarity and SBERT embeddings which can reduce the manual effort on tariff table alignment(Bayona; 2025). They were able to achieve an impressive level of accuracy approximately 90% with SBERT and 80% to 85% accuracy with TF-IDF cosine similarity which proved that text embeddings perform better results (Bayona; 2025). However, they are trying to achieve mapping between HS code to HS code from 2017 to 2020 and the description of the HS codes within a region hardly ever change, which is why they can get a strong result. Their methodology does not yield promising results under generalization conditions. In contrast, our research demonstrates through strong validations that the proposed approach can perform well under generalization conditions. Our research also uses advanced text embedding from OpenAI compared to the SBERT text embedding that was used for their research.

A study conducted by Otmane Amel *et al* is directly relevant to our research. In their study they have adopted a Multimodal Approach by providing Images of the products and limited text description of 2144 products with 16 distinct HS Codes to assign the correct Codes using pretrained models such as RestNet, Vit for image and SimCSE text encoder after removing punctuations and numbers(Amel et al.; 2023). They have stated that their study yielded the following metric results of 93.5% and 98.2% as their Top-3 and Top-5 accuracies respectively however the dataset they have done their research is significantly low, since there are more than 5000 HS codes present in real world. They also removed the numerical information within the product description which is not something reflected in real world scenarios. In contrast, our research provides insight into different sets of metrics such as Top-3, Top-7, Top-14, F1, Recall, Precision and we have explored encoding the hierarchical structure along with the description of the product without removing any numeric values. Even though their work considers realistic aspects of data ingestion, their methodology and model performance are limited to small-scale implementation.

By reviewing these related works published we identified several gaps that can be filled with our research such as validation for generalization, a wider collection of metrics to showcase performance and utilizing powerful embeddings to store descriptions. Most of the previous research was conducted with Machine Learning/ Deep Learning models to assign HS code classification but the drawback is they use one dataset to test, train and validate, but the dataset description follows the same linguistic structure which introduces a bias. Since their methodology has not been tested with a different dataset their proposed method struggles to materialize in real world. In contrast, we have performed testing different datasets in our research across different methodologies and provided a comparative analysis for the performance of our methodology.

2.3 RAG Methodology

We determined that traditional machine learning/ deep learning methodology are not performing at a level to implement in real world, hence we explored recent publications on text classification, and we identified promising work done using RAG (Retrieval Augmented Generation) and LLM (Large Language Models). One of the previous works that piqued our interest was published by Nathaniel *et al* where they have done a comparative analysis focused on In-Context learning between RAG combined with LLM and traditional machine learning models KNN and logistic regression(Huber-Fliflet et al.; 2024). Their work compared the precision and recall curve between zero-shot, few-shot and 3-shot prompting and concluded that the 3-shot prompting performs better when there are plenty of simple high-quality data is available but the zero-shot prompting gave better results when there aren't enough high-quality data(Huber-Fliflet et al.; 2024). They also stated that when the retrieval had too many dissimilar examples the Few-shot prompting performed poorly(Huber-Fliflet et al.; 2024). For our RAG and LLM implementation we have used Few-Shot prompting since we are retrieving the HS codes based on semantics, we will get similar results only and reduce the prompt length to the LLM, which will perform better than zero-shot and 3-shot prompting in-terms of response time. Their work has several limitations, since it's focused mainly on Binary Classification whereas our work is introducing multi-label classification, they have used RAG only to retrieve example whereas our RAG pipeline performs comprehensive knowledge base retrieval. Like many other previous works their methodology lacks validation for generalization.

The study conducted by Muhammad Arslana *et al* provided a descriptive information on several application of RAG in different domains and their performance evaluations. They have surveyed articles and categorized them into task-specific and discipline specific to showcase that the RAG application with LLM is still not applied widely enough to make progress and overly focused on potential of its application rather than the actual application(Arslan et al.; 2024). Even though their work provides us several information on RAG applications, it does not discuss the domain constraints, effectiveness and limitation of RAG and does not present a strong technical analysis. In contrast, our research fills this gap by providing an in-depth technical analysis and addresses the limitations of RAG with LLM architecture.

By surmising the previous works related to text classification, we decided to incorporate both RAG and RAG combined with LLM for our comparative analysis with Agentic AI framework. This will provide a clear understanding of the performance by the Agentic AI compared to the RAG architecture. Our findings will provide valuable insight for future research to explore the application of Agentic AI for Tax compliance.

2.4 Agentic AI

One of the latest technological applications that has gained significant attention in recent time is the Agentic AI Framework and its capability of automating workflow can be used for tax compliant. A recent publication by Pramod Kumar Siva explored the application of Agentic AI and the ethical concerns surrounding AI-driven tax compliance systems where they precent several evidence that many government entities have started to transition from plain LLM to Agentic AI to achieve autonomous workflow to reduce manual oversight, enforce consistent decision-making and process large dataset(Siva; 2025). Their work focuses the policy aspect of AI application for tax compliance and does not contain any technical information that can be used to achieve a transparent and auditable system for

tax compliance. In contrast, our research explores both policy and technical aspects of Agentic AI applications on HS Code assignment with comparative analysis with RAG and LLM to give a strong technical insight on the performance difference.

The application of Agentic AI is also gaining significant traction in financial sector, a study conducted by Mike Chen on Asset Management and Agentic AI has stated that ignoring integration of Agentic AI is not optional since it has the potential to change all aspects of asset management including risk management, compliance and servicing of clients(Chen et al.; 2021). They have proposed a multi-agent architecture for the exemplary scenario for an asset management workflow since concise modularized agents will perform better than a large collection of LLM. By basing this idea our project also follows multi-agent to classify the HS codes. Their research does not have a materialized implementation but rather a theoretical one, but we have used their idea and materialized it for our HS Code classification.

By comparing and critically reviewing related prior works we have established that the state of our work is an improvement upon their work and contributions. The novelty of our work is that we are comparing RAG, RAG with LLM and Agentic AI for HS code classification to justify integration of Agentic AI for Tax compliance. We have adopted and integrated similar techniques and evaluations that was used in the many previous publications and combined them to present our novel implantation for HS code classification based on product description. Additionally, our methodology performance is evaluated using different metrics an datasets (generalized condition) which was lacking in previous reserach and shared the findings for future research to extend upon our work.

3 Methodology

3.1 Introduction

The experimental design of this study is structured around the primary research objective of assessing whether Agentic reasoning provides measurable performance benefits beyond retrieval-based approaches for HS code classification. All methodological choices, including retrieval architecture, retrieval size, dataset selection, and evaluation metrics, were selected to systematically isolate the marginal contribution of Agentic AI while controlling for retrieval quality and linguistic variability. For our research we first identified core factor that is required for classification of HS codes based on description, which is capturing the semantics of the description and mapping the relevant HS codes. We decided to perform a combined approach of empirical and experimental to address the research questions by performing comparative analysis with different methodology such as RAG, RAG with LLM and Agentic AI using several evaluation metrics. Our research also validates if the proposed method performs well under generalization condition. Several previous works are used as a foundation to implement the project and among them most have used neural networks and text-embedders such as BERT or SBERT as their core architectural components, but the results yielded are underwhelming or they perform well only under a certain conditional setup. When it comes to legal texts there are several contexts with similar semantics which can be challenging to classify with aforementioned methods. Since these problems require a well distributed semantics and retrievals we choose RAG as the base of our methodology.

3.2 Retrieval Augmented Generation

According to a previous publication that used in-context learning and RAG stated that it improves the classification of legal texts (Huber-Fliflet et al.; 2024). The combination of RAG with LLM is another approach which hasn't received enough attention when it comes to tax compliance or HS codes classification(Arslan et al.; 2024). Since many taxes related sectors such as fintech are actively started to incorporate AI into their systems and Agentic AI is the current trend that show true potential for an autonomous tax compliance system (Siva; 2025). Using agentic AI to classify the HS code will be the novel approach of our methodology. To provide a progressive performance with three different methodologies RAG, RAG with LLM, Agentic AI we experimented with different method of storing data in vector database, different number of retrievals, different datasets and same evaluation metrics were used to measure the performance across each experiment.Figure 01 shows the architecture overview of our proposed methodology.

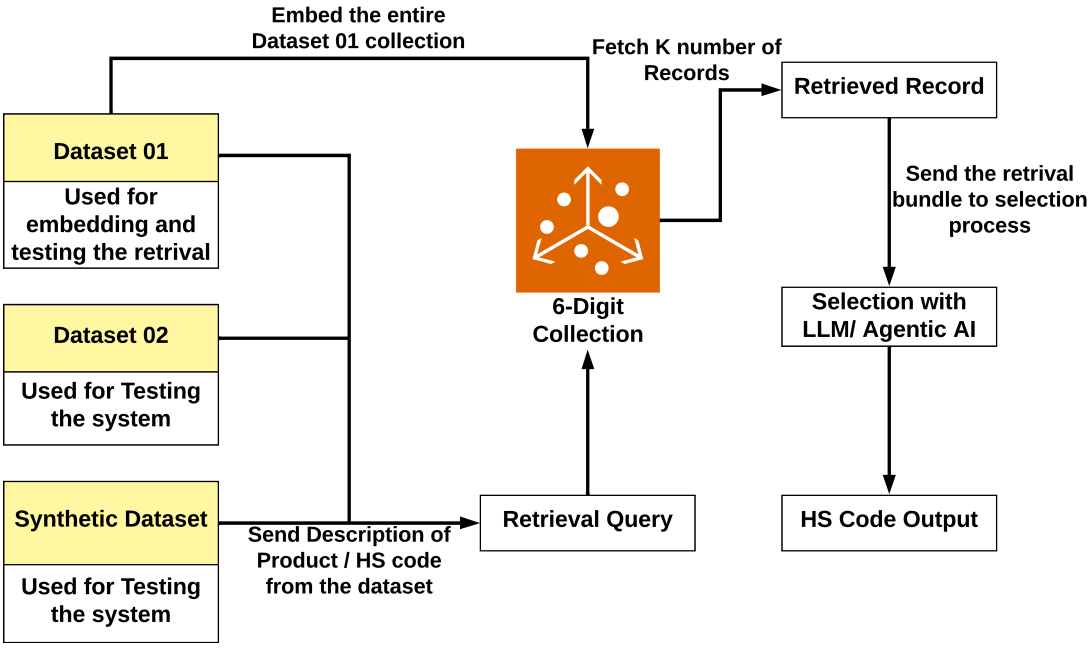

Figure 2: An Overview of core architecture

We obtained two set of datasets used in real world, one to embed into vector database (dataset 01) and another similar dataset to test the performance of the implementation under generalized conditions (dataset 02). These two datasets are similar in linguistic structure, where both datasets will have HS code and their corresponding description, but the text of that description may not be same, but they will point towards the same HS code. The following figure shows that for the same HS code there can different canonical descriptions but the semantic of these descriptions are same with respect to the import/ export domain.

We produced a third dataset that was synthetically generated for sales production description since there isn't any suitable dataset that fits the criteria for our research. Programmatically we selected 100 HS codes randomly and their respective descriptions to generate 300 synthetic product descriptions to test the implementation with a dataset that are not similar to the linguistic structure of the embedded dataset. We cleaned dataset 01 and 02 by standardizing the columns name, and the format for the HS Codes. We did not remove any numeric values from the descriptions and maintained the case of the

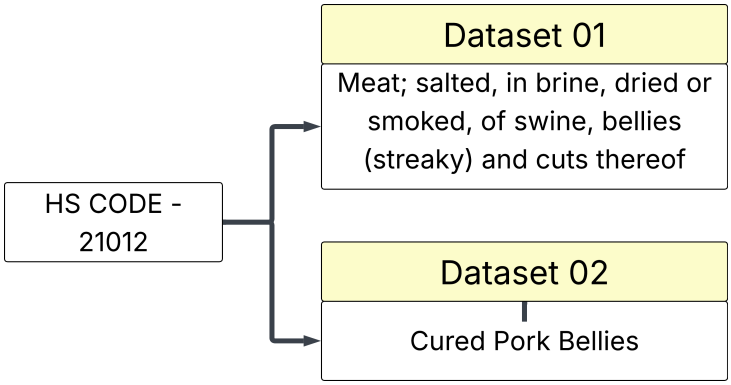

Figure 3: Dataset 01 and Dataset 02 HS Code description comparison

letters without any alterations. Next, Once the dataset is cleaned, we moved on to the process of embedding the descriptions and the HS codes into the vector database Chroma using Open AI embedding "text-embedding-3-large".We embedded the HS code and their descriptions from dataset 01 in 2 different ways, first we embedded the description without any changes (Flat HS Code), then we embedded the description by adding some hierarchical context into the description such as Section and Chapter highlighted by Blue and Red text respectively as shown in Figure 04.

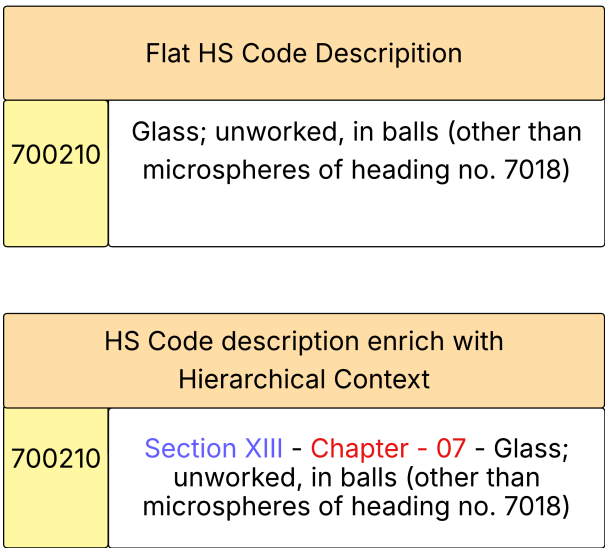

Figure 4: Hierarchical context enriched description

Once the embeddings are stored, we tested the RAG and RAG with LLM with 1000 HS codes from dataset 01, dataset 02 and then 300 products descriptions that were generated synthetically. To measure and compare the performance of both methodology we used Top-1 accuracy, Top-K accuracy, precession, recall and f1 as the metrics. The detailed explanation of these evaluations is provided under the Evaluation Section. We also experimented by changing the number of retrievals for both methodology and then we compared the performance between the two methodologies. This is because the quality of the retrieval documents is influence by the number of retrieval, if a low number of retrievals is used then the data will be narrowed where the correct document might not be there and a high number of retrieval will result in redundant documents being present in the retrieved data(Perçin et al.; 2025). We wanted to determine the optimal number of

retrievals that can be used for our RAG, RAG with LLM and Agentic AI methodologies to get the best performance.

3.3 Agentic AI

After experimenting with RAG and RAG with LLM implementation, we progressed to implementing Agentic AI. We integrated the Agentic AI framework into the RAG flow to feed the agents with concentrated data so that it can determine the best HS code and also reduce the chance of Hallucinated outcomes. We determined that the embedding without hierarchical context performed well for RAG based on the tests that were carried during our RAG and RAG with LLM experiments, hence we have used the same non-hierarchical collection for our Retrieval Agentic AI implementation. Multi-Agent RAG improves the overall quality of retrievals and the reliability of the response using several LLM agents to filter the noise and adjust based on the distribution score of the retrieval(Perçin et al.; 2025). Hence, we implemented Multi-Agent RAG by introducing reasoning agent and reflection agent. We also tested the implementation with 2 different approaches of retrieving records, flat retrieval (Figure 05) and hierarchical retrieval (Figure 06). In the study published by Shubham *et al* they trained a hierarchical classification model by training it at 2-digit, 4-digit and 6-digit division and it performed better than a flat model approach(Shubham et al.; 2023). Since we are basing our research upon their work, we designed the additional architecture that embeds the HS code and the descriptions in 3 separate collections, 2-digit, 4-digit and 6-digt respectively. By comparing these two designs we determine if storing the HS code with hierarchical structure will help improve the performance or not.

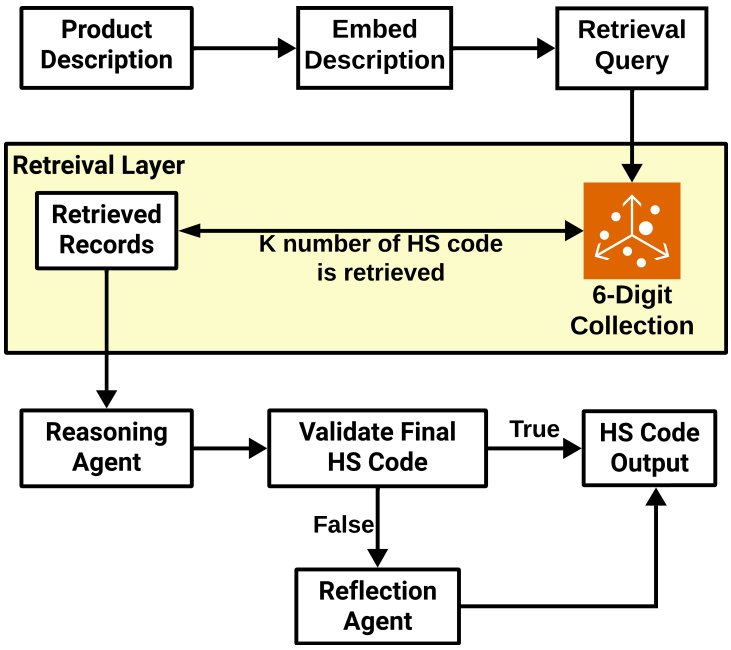

Figure 5: Agentic AI Flat Retrieval Architecture

When a product description is sent to the system we first embed and query the Chroma DB to retrieve the records that match in similarity. When it comes to the Flat HS code collection (Figure 5), we retrieve the K number of records and send it to the reasoning Agent to select the suitable HS code. However, in Hierarchical Retrieval Architecture

(Figure 06) we start the retrieval at 6-digits since the description of HS code at 2-digit and 4-digits are vague the chance of misclassification is high. Hence, we will use the information of 2-digit and 4-digit to provide better context and to validate the 6-digit for the reasoning Agent. For both Flat and Hierarchical retrieval only the retrieval layer components are different, the rest of the flow are same in both architectures. If the HS code selected by the reasoning agent is invalid, then the refection agent will be triggered and reevaluate the set of candidates that was retrieved. It will select another candidate from the retrieval records that matches with the semantics of the inserted description. If it's not able to select an alternative candidate, it will fall back to the initial HS code that was selected by reasoning agent.

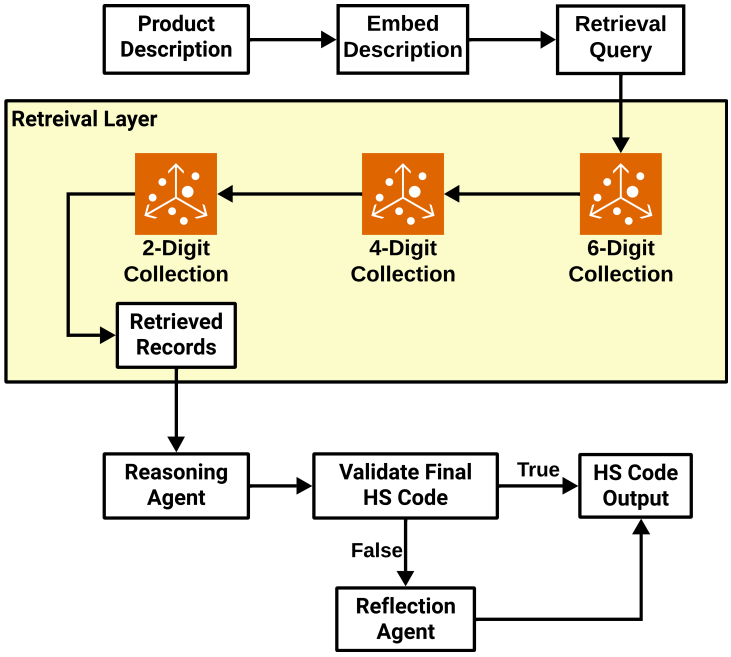

Figure 6: Agentic AI Hierarchical Retrieval Architecture

4 Design Specification

The final architectural design we created adopts a Hybrid approach of RAG combining Agentic AI supported by flat retrieval using Open AI embedding and ChromaDB. Unlike traditional Machine learning approaches, the design relies on semantics similarity search supported with LLM reasoning to determine the most suitable HS code for a given product description. The design processes unstructured product descriptions to predict the correct or the most suitable HS code. The design has 3 layers, Embedder layer, Retrieval Layer and then Agentic Layer as shown in the Figure 03,

4.1 Embedder Layer

In this layer the product descriptions are embedded using Open AI embedder "text-embedding-3-large" since it has 1536-dimentional vector representation which is higher than BERT or SBERT and then stored in ChromaDB. The entire dataset 01 is stored into the ChromaDB without any hierarchical context or structures. When retrieval query is

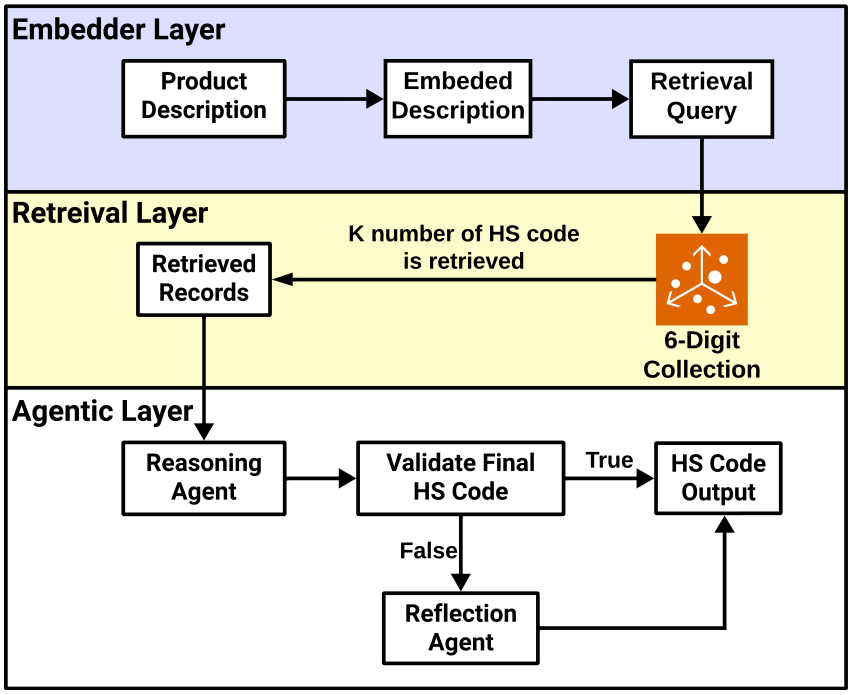

Figure 7: Layered Architecture Overview

sent to the ChromaDB, the records will be fetched based on Cosine distance corresponding to the product description we pass into the system

4.2 Retrieval Layer

The RAG Component will retrieve the records from flat HS code collection. In our RAG design we are intentionally avoid performing semantic validation right after the retrieval, instead we pass it to the agent for contextual validation which will give agent a broader set of records to access and determine the right HS code. Each test product description will be embedded and then sent to ChromaDB to fetch the similar set of descriptions stored in it and then it's sent to the AI Agents to select the most suitable HS code.

4.3 Agentic Layer

We implemented two Agents, a reasoning agent and a reflection agent. After the collection of HS codes are retrieved from ChromaDB the reasoning agent will access the collection and select the suitable HS code for the given description. Then the selected code will be sent to the reflection agent to validate the HS code and if the confidence score of the selected HS code is greater than or equal to 0.75 then we finalize the HS code without sending it to reflection agent. If the confidence score is low, then we send the selected HS code along with all the meta data and parent HS code information to the reflection agent to validate and reflect on the set of data to choose a better candidate. We only loop the reflection agent once, since multiple looping can cause high latency and consume the purchased API tokens. The reflection agent will look for the Top-k valid HS code candidate and the product description and if it's able to identify a better HS code, then it sends back to reconsider as the final output. If the outcome is invalid, then the fallback logic will be triggered where the previous output will become the final output. The reflection

agent will help reduce Hallucination from the LLM, check for mismatched hierarchies and low-confidence misclassification.

5 Implementation

This section focuses exclusively on the implementation of final hybrid design combining RAG and Agentic AI. The early implementation of RAG and RAG with LLM is not discussed since it was used for baseline validation and for comparative analysis. The section contains discussion on the design components developed, outputs generated and the tools used for the implementation.

5.1 Data Preparation and Embedding Layer

After obtaining datasets 01 and 02, we performed some data cleaning, first by standardizing the column name by making the HS code column and the HS code description column follow the same format and name in both datasets so that our system can process them same way. Both these datasets had additional column, but we removed them since they did not require our research. We did not remove any numerical value, or the punctuations present in the description since removal of these values does not reliably contribute to the improvement of the performance(Siino et al.; 2024). The OpenAI models are trained on raw texts and removing the punctuation will remove the context implied by the Canonical description. The descriptions are embedded without removing the punctions and spaces which can improve the embedding quality by capturing the syntactic context and word alignment implied within the HS code descriptions(Yuan et al.; 2020). Both datasets contained more than 5000 HS codes with similar canonical descriptions of the HS codes which is perfect to validate the methodology under generalized condition. Since the commercial datasets are not publicly available to use and many platforms do not store the HS codes, we had to create a synthetic dataset for the product with HS Codes.

The synthetic dataset was created by selecting 100 random HS code from the dataset 01 and used "gpt-4o-mini" with temperature 0.6 to generate 3 different sales product descriptions for each HS code based on the canonical description from dataset 01. The generated dataset had varied in description length, implicit forms and sales-style descriptions that are used by humans. It should be noted that the descriptions were not vague, however the linguistic structure was different compared to the canonical description of dataset 01 and 02. The reason for such dataset is in real world the goods that are getting traded have description similar human language rather than the formal or canonical structure that are being used in HS Code descriptions. It is quite challenging to determine the HS code at that level and human involvement is required to further process the product.

Once the datasets are cleaned, we start embedding the HS code descriptions from dataset 01 into the chroma DB using OpenAI embedded "text-embedding-3-large". The product description from dataset 01 are stored in two different ways. One approach is flat HS code retrieval where it is not stored in hierarchical structure and the other approach, Hierarchical retrieval which is creating three separate collection to store Chapter, Heading and HS code as 2-digit, 4-digit and 6-digit respectively in order to mimic the hierarchical nature of the HS code which was done in a prior research. We stored the HS code in the following collections.

- Flat HS Code – 6-digit (hs_codes)

- Chapter Level – 2-Digit (hs_codes_2d)
- Heading Level – 4-Digit (hs_codes_4d)
- HS-Code Level – 6-digit (hs_codes_6d)

When storing the data we mapped the information as follows for both Flat and Hierarchical retrieval:

- Id – HS Codes
- Documents – HS Code description
- Metadata – Description, Level, Parent Code

After embedding all the HS codes and their canonical description in the ChromaDB we have established a queryable vector knowledge base that supports both RAG and Agentic workflow.

5.2 Retrieval Layer Implementation

When the product description is given to the system it will first embed the description and then query the ChromaDB and fetch the K number of records using cosine distance. The cosine distance is used to calculate confidence score of the HS code using following formula,

- Confidence Score = 1 – Cosine Distance

This allows us to interpret the retrieval results consistently throughout RAG and Agentic AI pipeline by maintaining a normalized confidence score where higher value means greater semantic similarity. Our experiment on RAG and RAG with LLM using different vector collections proved that the flat embedding resulted in better performance than the collection that was created by including hierarchical information into the description of the HS Code. Hence, for our multi-agent RAG architecture we decided to use Flat HS code collection.

We created two RAG pipelines where the flat retrieval consists of the complete HS codes and their canonical description, and the hierarchical retrieval stores the HS codes in separately and retrieves in a 6-4-2 HS code sequential order. By performing multiple tests and evaluation it was decided the flat retrieval architecture performed better than the hierarchically structure retrieval. The retrievals from these are sent to the Agents to assess and select suitable HS code among the retrieval bundles.

5.3 Agentic Layer Implementation

The Agentic layer is primarily responsible for reasoning and reflection of the retrieved HS codes from the ChromaDB. Both agents are implemented using OpenAI model "gpt-4o-mini" with temperature 0.0 for the reasoning agent and 0.2 for reflection agent. We set the temperature low since our implementation require more focused and deterministic outcome from the agents. For both Agentic implementation we have utilized Zero-shot prompting, since we don't have variety of High-quality HS code description or the Production description, based on the statement from previous research by Nathaniel *et al*

the zero-shot prompting work better when data quality is low or limiting when compared to 3-shot prompting(Huber-Fliflet et al.; 2024). Another reason for using Zero-shot prompting is HS code information that we pass per prompt is around 30 HS code descriptions, along with metadata and parent HS code information for context. The prompt size or the number of tokens assigned to a prompt has significant impact on performance and output, a large amount of information will impact the model's input-context, and it will not be able to access and use the information robustly(Liu et al.; 2024). We determined the number of HS codes to provide for the agent as 20 based on the experiments we performed with number of retrievals for the RAG and RAG with LLM. Since the final output is given by the LLM agent, there are possibilities of invalid outputs that can impact the evaluation process, which is why we added a fallback logic. The fallback logic is triggered when the LLM outputs None, empty, incorrect format, non-numeric values, this ensures that the output is always numeric and does not impact the evaluation. The confidence score is set to 0.0 when it's an invalid output. The predicted HS codes and the metadata are stored in CSV and JSON files for record keeping purposes and use it to analyze for the improvement of future works from the project. The entire process is evaluated with Top-1, Top-K, precision, recall, f1 and latency and the detailed information regarding these experiments is given in detail under Evaluation section below.

The below table contains the list of tools and techniques used for the implementation of the proposed system.

Table 1: Tools and technologies used in the research

Type	**Tools & Techniques**
Programming Language	Python 3.10
AI / LLM Models	OpenAI GPT-4o-mini (reasoning), OpenAI `text-embedding-3-large` (embeddings)
Vector Database	ChromaDB (PersistentClient), Cosine similarity search, Hierarchical collections: `hs_codes_2d`, `hs_codes_4d`, `hs_codes_6d`
RAG Framework Components	Semantic retrieval, Top-K similarity search, Zero-shot prompting, Hierarchical 6→4→2 retrieval, Reflection-based agentic reasoning, Fallback logic
Data Processing	Pandas, NumPy, JSON, CSV, Data cleaning and normalisation
Machine Learning Evaluation	Scikit-Learn (accuracy, precision, recall, F1), Latency measurement, JSONL logging, Experiment tracking
Development Environment	Visual Studio Code, Jupyter Notebook, Anaconda (virtual environments), Python virtual environment management
Infrastructure & APIs	OpenAI API (embeddings + reasoning), Local storage for vector DB, Environment variable management (.env)
Support Tools	TQDM (progress bars), Lucid Charts (architecture diagrams), Git (version control)

6 Evaluation

We performed several experiments on base RAG, RAG with LLM and Agentic AI. This section we will not cover all the experiments instead it highlights only those whose results directly informed and shaped the final methodology and contributed to addressing the research questions. All the experiments were evaluated using Top-1, Top-K, Precession, Recall and F1 and three methodologies were tested using all three datasets (dataset 01, dataset 02, synthetic dataset). For RAG, RAG with LLM and Agentic AI we have used 6 use cases to test the performance and to simulate the generalized conditions by testing the methodology using dataset 02 and synthetic dataset. The in-depth analysis for all experiments is visualized and discussed in the following sections.

Use Case List

- Case 01 (Dataset 01) - Descriptions are embedded without hierarchical context and evaluated using exact descriptions that were embedded to vector DB
- Case 02 (Dataset 01) - Descriptions are embedded with hierarchical context and evaluated using exact descriptions that were embedded to vector DB
- Case 03 (Dataset 02) - Dataset 02 is used for the evaluation for descriptions that are embedded without hierarchical context.
- Case 04 (Dataset 02) - Dataset 02 is used for the evaluation of descriptions that are embedded with hierarchical context.
- Case 05 (Synthetic) - Synthetic Dataset is used for the evaluation for descriptions that are embedded without hierarchical context.
- Case 06 (Synthetic) - Synthetic Dataset is used for the evaluation for descriptions that are embedded with hierarchical context.
-

6.1 Experiment 01 - Base RAG Method

This experiment is focused on analyzing the difference in performance when HS code is stored with and without hierarchical context. As mentioned under research methodology we created an additional collection where we add the hierarchical details into the HS code description of dataset 01 to enrich the description.

We tested all three datasets with 3 different K values which represent the number of retrievals from ChromaDB. Below visualization shows how the base RAG performance changes based on the quantity of retrievals and the way the HS descriptions are embedded into the ChromaDB.

The evaluation was conducted over six distinct use cases representing different types of product descriptions, including both real and synthetic datasets. The main goal of the evaluation is to determine if enriching HS code description improves the overall accuracy of HS code classification.

All the metric shows clear distinctions between all three datasets. Dataset 01 which contains the same descriptions as the ones we embedded into ChromaDB has the highest performance outcome; however, this is expected since we are evaluating if the embedded details are accurately retrievable or not. Our main analysis is between dataset 02

Table 2: Combined Evaluation Metrics for All Retrieval Settings

Metric	Case 01	Case 02	Case 03	Case 04	Case 05	Case 06
Retrievals Used: 03						
Number of Samples	1000	1000	1000	1000	300	300
Top-1 Accuracy	0.977	0.931	0.758	0.708	0.5667	0.5633
Top-3 Accuracy	1.000	0.992	0.933	0.903	0.780	0.7667
Precision	0.9550	0.8732	0.6147	0.5526	0.3886	0.3911
Recall	0.9550	0.8742	0.6239	0.5619	0.2936	0.2789
F1 Score	0.9550	0.8736	0.6177	0.5557	0.3218	0.3129
Retrievals Used: 07						
Number of Samples	1000	1000	1000	1000	300	300
Top-1 Accuracy	0.975	0.930	0.758	0.710	0.5667	0.5667
Top-7 Accuracy	1.000	0.989	0.969	0.951	0.8767	0.8700
Precision	0.9512	0.8711	0.6147	0.5547	0.3886	0.3930
Recall	0.9512	0.8716	0.6239	0.5639	0.2936	0.2819
F1 Score	0.9512	0.8713	0.6177	0.5577	0.3218	0.3154
Retrievals Used: 14						
Number of Samples	1000	1000	1000	1000	300	300
Top-1 Accuracy	0.976	0.933	0.758	0.707	0.570	0.560
Top-14 Accuracy	1.000	0.993	0.801	0.782	0.930	0.890
Precision	0.9531	0.8756	0.4382	0.4185	0.3750	0.2507
Recall	0.9531	0.8761	0.4490	0.4302	0.2729	0.1766
F1 Score	0.9531	0.8757	0.4417	0.4223	0.3015	0.1986

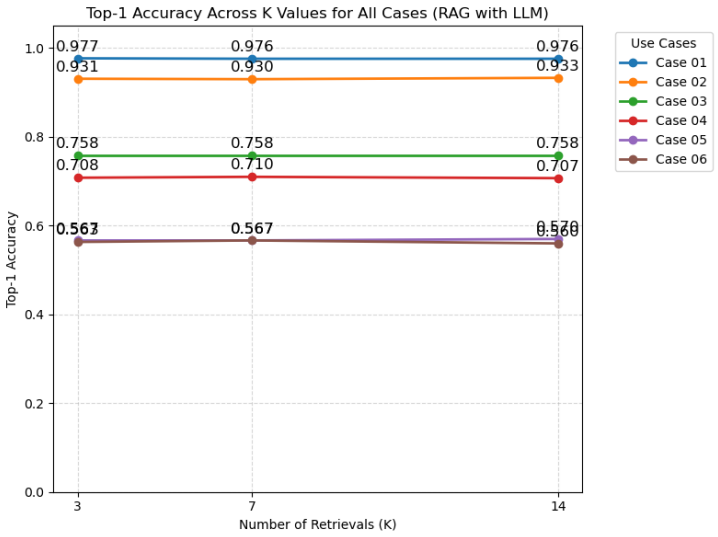

Figure 8: Base RAG Multi-Line Graph for Top-1 Accuracy between different K values

(Case 03,04) and synthetic dataset (Case 05,06) where we can see a significant gap in performance across all metrics. As mentioned before, the dataset 02 contains similar canonical description to dataset 01, which is why the performance is better than the synthetic dataset that contains descriptions in the format of sales description of the products. The visualization from Figure 8 indicates that the Top-1 accuracy for Case 03 and 04 is approximately 0.758 and 0.708 at K=3 and 0.758 and 0.707 at K=14 respectively.

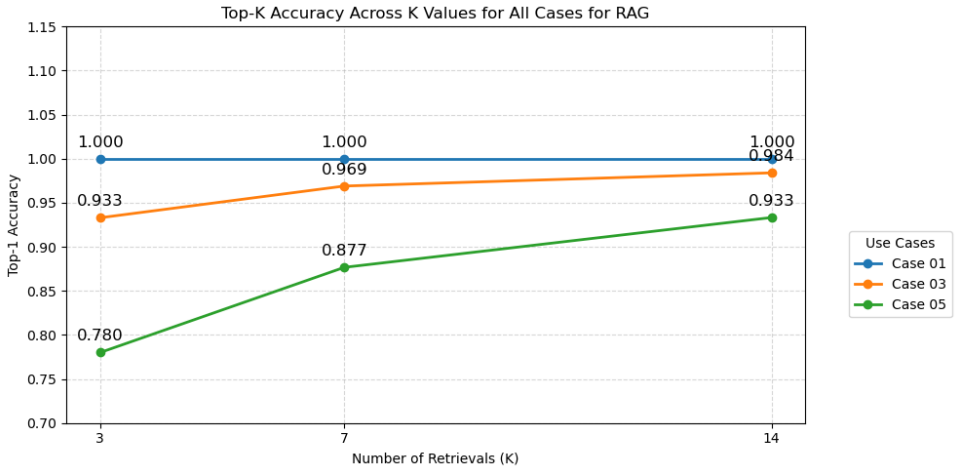

Figure 9: Base RAG Top-K accuracy comparison for different K values

Case 05 and Case 06 with Top-1 accuracy approximately at 0.5667 and 0.5633 at K=3 and 0.57 and 0.56 at K=14 which also indicate similar pattern where the Top-1 accuracy is higher when performing retrieval from non-hierarchical context collection. This clearly proves that enriching HS Code with Hierarchical context does not improve the accuracy.

For case 03 the Top-1 accuracy remains the same approximately at 0.758 at K=3, 7, 10 which indicates there is no impact in accuracy when increasing K. However, when comparing the Top-K accuracy in Figure 09 we can witness an increase among retrieval accuracy for Case 03 and Case 05 approximately from 0.93 at K=3 to 0.98 at K=14 which indicates increasing the K improves the chances of getting the correct HS code, however, it does not guarantee the correct classification. The same pattern of increase in Top-K can be observed for Case 04 and Case 06 from Table 2 but the values are lower than case 03 and 05 which indicates enriching the description does not outperform flat HS code. Based on these results we discarded the hierarchical enrichment experiment from this point onwards since the quality of retrieval impacts the performance of the overall outcomes involving Case 02, Case 04 and Case 06 will not contribute to any further insights.

6.2 Experiment 02 - RAG with LLM Method

This experiment was performed to assess the impact in accuracy, precision, recall and F1 when base RAG is integrated to an LLM. Findings from this experiment will also help us prove the need for Agentic AI.

The evaluation was conducted only for Case 01, 03, 05 since in the first experiment we determined that enriching hierarchical context does not increase the accuracy or any other metrics. The Top-1 accuracy for Case 03 and 05 is approximately 0.83 and 0.6733 at K=3 and 0.845 and 0.7133 at K=14 respectively. As seen in Figure 11 this indicates the Top-1 accuracy increases when we combine an LLM to assess and select a suitable HS Code.

When we compared the Top-1 accuracy with Top-K accuracy for Case 03 and Case 05 we could observe in Table 3 at K=3 the Top-K are 0.933 and 0.78 and at k=14 the Top-K are 0.985 and 0.9333 respectively. We can observe that the Top-K accuracy is higher than the Top-1 Accuracy which means the retrieval bundle contains the correct HS code, however, a basic LLM integration still struggles to select the correct HS code. This strongly suggests that we can increase the accuracy of the classification by integrating the RAG with Agentic AI. Unlike the Base RAG model, whose gains stagnates for synthetic

Table 3: Evaluation Metrics Across Retrieval Settings for Selected Use Cases

Metric / Case	Case 01	Case 03	Case 05
Number of Retrievals = 3			
Number of Samples	1000	1000	300
Top-1 Accuracy	0.977	0.83	0.6733
Top-3 Accuracy	1.000	0.933	0.78
Precision	0.955	0.714	0.4807
Recall	0.955	0.7186	0.372
F1 Score	0.955	0.7154	0.4055
Number of Retrievals = 7			
Number of Samples	1000	1000	300
Top-1 Accuracy	0.977	0.839	0.71
Top-7 Accuracy	1.000	0.969	0.8767
Precision	0.955	0.7259	0.5119
Recall	0.955	0.7308	0.4226
F1 Score	0.955	0.7275	0.4524
Number of Retrievals = 14			
Number of Samples	1000	1000	300
Top-1 Accuracy	0.98	0.845	0.7133
Top-14 Accuracy	1.000	0.985	0.9333
Precision	0.9608	0.7348	0.5148
Recall	0.9608	0.7393	0.4221
F1 Score	0.9608	0.7362	0.4527

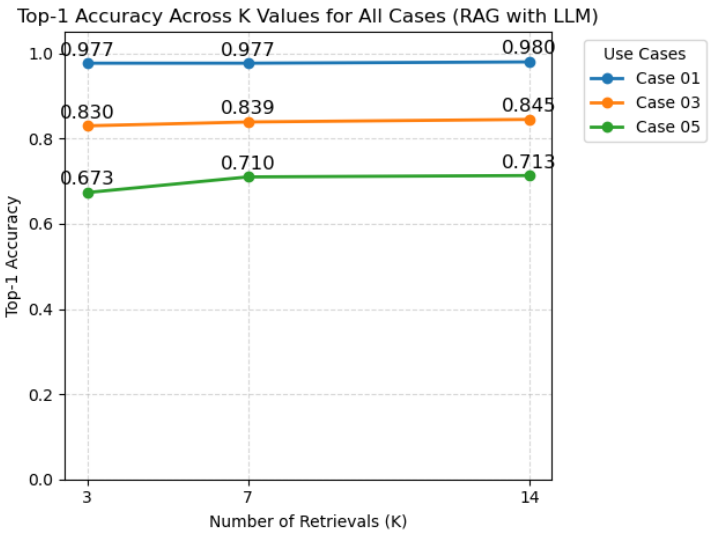

Figure 10: RAG + LLM Multi-Line Graph for Top-1 Accuracy between different K values

dataset, the RAG combined LLM system effectively utilizes higher retrieval sets to enhance both precision and recall. The most significant performance gains are observed at K = 14, where the LLM benefits from richer contextual evidence, achieving near-perfect Top-K accuracy and high Top-1 accuracy in several cases. These results reinforce the value of combining retrieval with LLM reasoning in HS code classification and highlight that LLM-based re-ranking and decision-making play a central role in managing noise

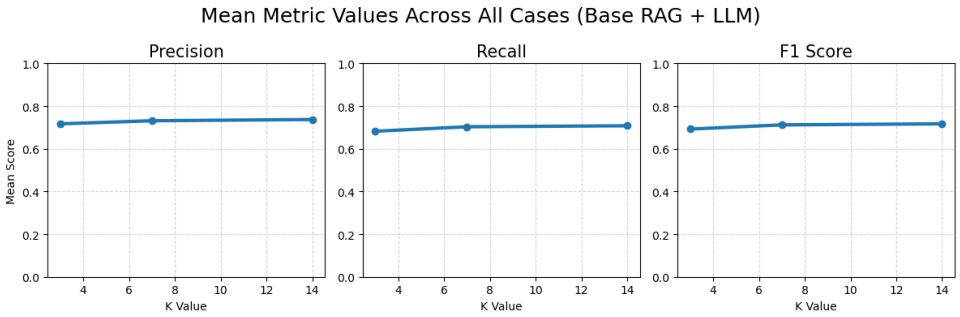

Figure 11: RAG + LLM Subplot Line Graph for Precision, Recall and F1 for different K values

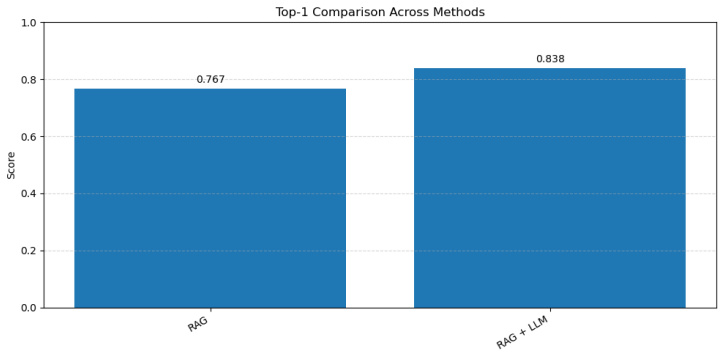

Figure 12: RAG vs RAG + LLM Top-1 Accuracy performance

introduced by larger retrieval sets.

6.3 Experiment 03 – Agentic AI with Flat Retrieval

This is our main experiment where we integrate Agentic AI into the retrieval flow, or more specifically we convert the integrated LLM to Agentic AI to assess and select the correct HS code with reasoning and reflection of the choice. We will exclude Case 01(dataset 01) for the following experiments since the performance from RAG and RAG with LLM yielded high accuracy and it was used as a baseline to check for performance margin in previous experiments.

Table 4: Agentic AI Evaluation Metrics for Use Cases 03 and 05

Metric / Case	**Case 03**	**Case 05**
Number of Samples	1000	300
Top-1 Accuracy	0.9060	0.7367
Top-5 Accuracy	0.6910	0.6620
Precision	0.8336	0.5427
Recall	0.8389	0.4492
F1 Score	0.8353	0.4799
Total Eval Time (sec)	2207.15	709.00
Avg Latency (sec/sample)	2.2072	2.3633

After integrating agentic AI, the HS code classification for Case 03 (External Dataset)

and Case 05(synthetic dataset) increase significantly. We set K=20 since our previous experiments proved that by increasing the number of retrievals the chances of fetching the correct HS code increases. Case 02 achieved Top-1 Accuracy approximately 0.906 with external description similar in linguistic structure whereas Case 05 achieved 0.7367 on synthetic description which is 22.98% lower than Case 05. The reason for such a gap between two cases is the synthetic dataset has sales like descriptions which are vague and does not follow the canonical structure that are used in HS code systems. This proves linguistic structure should be considered while embedding the details into the vector DB.

Other metrics, precision, recall, F1-score also incresaed substantially for synthetic dataset, while Top-20 retrieval values are closer for both Case 03 and Case 05, the AI agents could not assess and classify the correct HS codes for the synthetic dataset.The average latency to process the description for Case 03 and Case 05 is almost same which proves that computation cost is not impacting the performance of the designed model. The performance of the designed model is impacted by the semantic quality of the description of the product passed. Hence, even if we improve the prompt or increase the number of retrievals, if the description passed is vague, the agents will struggle to classify the HS code.

6.4 Experiment 04 – Agentic AI with Hierarchical Retrieval

Prior research motivated us to check if the Agentic AI yields better result by storing the HS code Hierarchically and then retrieve them in a bottom-up sequence and make comparison with the Flat retrieval approach we designed in Experiment 03.

Table 5: Hierarchical Retrieval Agent Evaluation Metrics for Case 03 and 05

Metric / Case	**Case 03**	**Case 05**
Number of Samples	1000	300
Top-1 Accuracy	0.661	0.595
Top-5 Accuracy	0.776	0.662
Precision	0.5192	0.4282
Recall	0.5085	0.4365
F1 Score	0.3827	0.4309
Total Eval Time (sec)	6058.71	2109.00
Avg Latency (sec/sample)	6.0587	6.0633

The hierarchical retrieval design performs worse than the flat retrieval by achieving a Top-1 accuracy approximately 0.661 for Case 03 and 0.595 of Case 05. Values of other metrics also decreased significantly for this design reflecting instability in classification. The average latency of this design is 6 seconds higher than flat retrieval which indicates this methodology requires high computation power. The reason for such poor performance is even if the correct 6-digit HS code is fetched, the agents might interpret it differently if the 2-digit or 4 digit retrieved collections are incorrect. This will contradict and confuse the agents while assigning the HS code. The hierarchical retrieval may mimic the real hierarchical representation of HS code; however, implementation of such structure does not contribute to the performance of the Multi-agent RAG implementation. This type of approach appears to be more effective when it comes to Deep Learning models which were used in the research done by Subham *et al.*

6.5 Discussion

The research evaluates RAG, RAG with LLM and Agentic AI pipelines for Harmonized system code classification with different cases, dataset, number of retrieval and storage structure (Flat vs Hierarchical). With respect to RQ2, the results indicate that Agentic reasoning improves Top-1 accuracy and precision primarily when the correct HS code is already present within the retrieved candidate set, suggesting that reasoning quality, rather than retrieval alone, drives the observed performance gains. These findings reveal how retrieval architecture, data quality, and Agentic reasoning jointly impact classification accuracy and computational cost. This section will discuss these results in detail and compare them with existing findings from related works.

Findings related to RQ3 demonstrate that linguistic mismatch between canonical HS descriptions and sales-style product descriptions remains a dominant limiting factor across all evaluated architectures, including Agentic AI, despite improvements in reasoning and reflection. Deriving the correct HS code from sales-style product descriptions remains challenging when the system is optimised for canonical HS descriptions. The generated synthetic dataset may not be up to the standard of commercially available data and also synthetically generated data may poorly mimic human-like textual representations, which negatively affects classification performance (Li et al.; 2023). This explains why the metrics values for synthetic datasets are lower than canonical dataset 02. However, in real world there is a gap between the description of the product and the description used for HS code classification. A form of textual conversion and domain expertise is required to map them correctly.

A significant outcome of our experiment on Flat vs Hierarchical retrieval revealed that the Flat retrieval outperforms the Hierarchical retrieval even though the Hierarchical retrieval reflects the structure of HS Code, its performance is 37.06% lower than the Flat retrieval for dataset 02 and 23.85% lower for synthetic dataset. A similar observation was made in a previous study by Lewis *et al* where they warn that this type of retrieval can amplify noises and lead to mistakes at one hierarchy level and that leads to misclassification the next hierarchical level(Lewis et al.; 2020). Another information from our study is that these HS code hierarchical structures were created with legal context, and the legal hierarchical context cannot be reliably captured through semantic similarity alone. For example, two HS codes can be the same chapter with matching semantics, however, the legal context represents them as a distinct description. This proves that the HS code hierarchy is not compatible with semantic embeddings and mirroring such structure leads to poor performance.

Researcher	**Method**	**Metric**	**Value**	**Dataset 02**	**Synthetic**
Shubham et al. (2023)	Flat Classification Model (6-digit level)	Accuracy	Flat: 56%	Flat Retrieval: 90.6%	Flat Retrieval: 73.7%
Shubham et al. (2023)	Hierarchical Classification (2+4+6 digit)	Accuracy	Hierarchical: 65%	Hierarchical Retrieval: 66.1%	Hierarchical Retrieval: 59.5%
Amel et al. (2023)	Multimodal Approach (Images + Text)	Top-3	93.5%	Dataset 02: 93.3%	Synthetic: 78%

Our proposed methodology outperforms the methodology proposed by Shubham *et al* and almost matches the results produced by Otmane Amel *et al*. It is important to note that our

methodology results were produced under a generalized condition by testing it with an external dataset that is similar in linguistic nature and synthetic dataset that is similar in sales description to simulate generalized conditions. This is not performed in previous research and by testing the methodology under a generalized condition we have filled the gaps in their research and proved that our methodology will yield better results than their methodology under generalized conditions.

Table 7: Comparison of RAG, RAG with LLM, and RAG with Agentic AI for Dataset 02

Metric	**RAG**	**RAG + LLM**	**Agentic AI**
Top-1 Accuracy	0.758	0.845	0.906
Precision	0.6147	0.7348	0.8336
Recall	0.6239	0.7393	0.8389
F1 Score	0.6177	0.7362	0.8353

Table 8: Comparison of RAG, RAG with LLM, and RAG with Agentic AI Synthetic Dataset

Metric	**RAG**	**RAG + LLM**	**Agentic AI**
Top-1 Accuracy	0.5700	0.7133	0.7367
Precision	0.3906	0.5148	0.5427
Recall	0.2969	0.4221	0.4492
F1 Score	0.3250	0.4527	0.4799

By comparing the performance of all three methodologies Table 7 and Table 8 with dataset 02 and synthetic dataset we can observe clearly that adding Agentic AI increase the HS code classification across all metrics. The accuracy increases by 8.7% when the LLM is integrated to the base RAG and when we integrate the Agentic workflow the accuracy increases by 19.53% from the base RAG and 6.1% over LLM. If we compare the precision base RAG produces more false positives and integrating LLM provides a way to understand context of the HS code description. However, after Agentic AI is integrated with the reasoning and reflection agent the precision increase significantly from 0.6147 to 0.8336. High precision is crucial for classifying HS code, since an incorrect classification can lead to incorrect tariff, legal issues and financial penalties. The recall also increases from 0.6239 to 0.8389 which again confirms that our methodology reduces the chance of missing the correct HS code classification and the methodology can perform well with diverse product descriptions that are canonically structured. In relation to RQ4, the observed improvements in precision and error reduction suggest that Agentic AI may support semi-automated HS code assignment in compliance-sensitive settings, rather than fully autonomous classification.

The synthetic dataset evaluation also follows the same incremental trend as dataset 02, however the overall metrics values are lower than the values that was produced from testing dataset 02. This proves that our methodology is sensitive to embedding quality and to the linguistic structure of the dataset. To mitigate this, a collaboration with domain experts is required to map and identify the structural differences between the product description used in real world and the canonical description of HS Codes. We have used multiple datasets that allowed us to make clear comparisons between external and synthetic datasets. We have explored different methodologies, RAG, RAG with LLM and Agentic AI and identified the limitations in each methodology. Our research has provided a wide range of evaluation metrics such as Top-K accuracy, precision, recall, F1 and latency and our experiments were conducted using realistic HS code dataset similar to prior work. The insights gained from these experiments have given

us a clear understanding of the potential the hybrid approach of combining RAG and Agentic AI in HS code classification and other tax code classification that are yet to be researched with our methodology.

7 Conclusion and Future Work

The goal of the research was to investigate the effectiveness of using Agentic AI to classify Harmonized System Codes based on the description of the products. This research is primarily centred on RQ2, which examines the marginal value of Agentic reasoning over retrieval-based methods for HS code classification under realistic linguistic conditions. The remaining research questions provide supporting context by addressing feasibility (RQ1), generalisation (RQ3), and compliance relevance (RQ4).

- RQ1 – Either by integrating an LLM with RAG or integrating Agentic AI with RAG the HS code classification can be done effectively with high performance across all metrics. This was confirmed by comparing the experiment results.
- RQ2 – Our experiments demonstrate that integrating an Agentic AI methodology yields consistently higher performance across Top-1, Top-K, Precision, Recall, and F1 metrics compared to base RAG and RAG with LLM.
- RQ3 – The Agentic AI methodology performed well under generalized condition by achieving 90.6% with external dataset and 73.7% with synthetic dataset. With Experiment 03 we were able to confirm these metrics' values.
- RQ4 – Based on prior research and our findings, Agentic AI has the potential to support AI-assisted systems for Customs Import/Export processes that are highly dependent on tax compliance and domain expertise.

Our findings show that Flat RAG with Agentic AI achieved overall Top-1 Accuracy of 90.6% for external canonical dataset and 73.7% for synthetic dataset which is higher than the results from Base RAG or RAG with LLM or Hierarchical retrieval methodologies. This confirms that RAG significantly improves classification performance compared to traditional ML and deep learning approaches, aligning with prior work showing that retrieval quality strongly influences LLM outputs. Across all experiments, dataset 02, which is similar to embedded data in ChromaDB performed better than the synthetically generated sales like product descriptions. This proves the sensitivity of embedding-based retrieval to the linguistic nature of input text. Overall, the research successfully attains its objectives by evaluating multiple methodologies with different numbers of retrievals and demonstrates advantages and disadvantages of Agentic reasoning in this domain. Our research also provides quantifiable evidence that HS code classification performance heavily depends on retrieval structure and input quality. The results suggest that the linguistic distinction between canonical HS code descriptions and sales-style product descriptions must be reduced to move towards more reliable semi-automated HS code assignment systems. In addition, the legal structure of the HS code hierarchy must be represented appropriately to support robust classification. A study conducted by Andrew Grainger supports the need for assisted systems to reduce the pressure and risk associated with classifying HS codes(Grainger; 2024). The methodology that we propose can enable a semi-automated system that can be used as an assistant to the customs officials to assign the HS codes by pre-classifying items with high confidence and reduce the manual processing time.

All research works has limitation and this study is no exception. The use of synthetic data for product description is a main limitation as the generated dataset lacks in depth, variety and a reflection of real-world text and this caused the low-quality retrieval bundle that we witnessed

during our experiment with synthetic dataset. Although the Agentic AI implementation yielded strong results, there are limitations associated with using the OpenAI endpoint. We were allowed to make only 10,000 API calls per day, which resulted in diminishing the number of samples and variation that we could test per day. Another limitation of our RAG implementation is the absence of cross-encoder re-rankers, which are widely used to stabilise retrieval ranking. which could have improved the Top-1 accuracy and Top-K accuracy. The final limitation of our research is the stochastic nature of the OpenAI model "gpt-4o-mini" that resulted in slightly different performance metrics plus or minus 1% difference every iteration we performed the experiments. Despite these limitations the research provides strong qualitative evidence that HS code classification can be semi-automated and has the potential to support more reliable semi-automated systems by addressing the limitations that were discussed before.

Since our research confirms that the Agentic AI can effectively classify HS code under generalized condition there are several avenues for future works that can be built based on our methodology and findings. One of the best practices used for RAG is utilizing Dense Hybrid Retrieval which can be used to reduce the retrieval ambiguity and improve recall. Another improvement that can be done is inclusion of a Cross-encoder re-ranking for final HS code selection. The future research can try to implement a different agent into the workflow to produce auditable reports that can be used by domain experts and customs officials to assign the HS code. We will also explore similar implementation for European Union Customs CN code classification and then calculate the relevant Tariff rates.

In conclusion the research showcases the performance difference between RAG, RAG with LLM and Agentic AI and proves that RAG with Agentic AI approach can effectively support HS code classification by achieving strong evaluation metric values for dataset with canonical description. While the performance gap between canonical description and synthetic dataset proves the need for a methodology to abridge the linguistic structural difference between such datasets. With further collaboration from domain experts and other researchers' the linguistic structural gap can be reduced, and more robust semi-automated HS code classification systems can be developed using the proposed methodology. The rapid advancement of AI and its application in tax compliance is growing and presents many opportunities to address gaps identified in previous research. The integration of Agentic AI and RAG has the potential to enhance regulatory alignment with efficiency, consistency and overall effectiveness in tax compliance processes. Overall, the findings indicate that while retrieval quality remains the dominant factor in HS code classification performance, Agentic reasoning contributes meaningful improvements in decision reliability when operating on linguistically consistent inputs.

References

Amel, o., Stassin, S., Mahmoudi, S. A. and Siebert, X. (2023). Multimodal approach for harmonized system code prediction, *ESANN 2023 proceesdings*, ESANN 2023, Ciaco - i6doc.com, p. 181–186.
URL: *http://dx.doi.org/10.14428/esann/2023.ES2023-165*

Arslan, M., Ghanem, H., Munawar, S. and Cruz, C. (2024). A survey on rag with llms, *Procedia Computer Science* **246**: 3781–3790. 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024).
URL: *https://www.sciencedirect.com/science/article/pii/S1877050924021860*

Bajpai, A. (2024). Evaluating the impact of artificial intelligence on enhancing tax compliance and financial regulation, *International Journal of Multidisciplinary Research and Technology* . Available at: www.ijmrtjournal.com.

Bayona, P. (2025). Beyond six digits: Automated tariff line hs transposition using natural language processing, *Technical report*, Economics Econstor.
URL: *https://www.econstor.eu/handle/10419/314422*

Belahouaoui, R. and Attak, E. H. (2024). Digital taxation, artificial intelligence and tax administration 3.0: Improving tax compliance behavior – a systematic literature review using textometry (2016–2023), *Accounting Research Journal* **37**(2): 172–191.

Chen, M. (2025). Agentic ai and the future of institutional asset management, *Journal of Portfolio Management* .

Chen, X., Bromuri, S. and Van Eekelen, M. (2021). Neural machine translation for harmonized system codes prediction, *ACM International Conference Proceeding Series*, pp. 158–163.

Dudu, O. F., Alao, O. B. and Alonge, E. O. (2024). Conceptual framework for ai-driven tax compliance in fintech ecosystems, *International Journal of Frontiers in Engineering and Technology Research* **7**(2): 001–010.

Ezeife, E. (2021). Ai-driven tax technology in the united states: A business analytics framework for compliance and efficiency, *International Journal of Multidisciplinary Research and Growth Evaluation* **2**(1): 693–701.

Grainger, A. (2024). Customs Tariff Classification and the Use of Assistive Technologies, *World Customs Journal* **18**(1): 3–32.

Huber-Fliflet, N. et al. (2024). Experimental study of in-context learning for text classification and its application to legal document review in construction delay disputes, *Proceedings of the 2024 IEEE International Conference on Big Data*, pp. 2119–2125.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S. and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks, *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Curran Associates Inc., Red Hook, NY, USA.

Li, Z., Zhu, H., Lu, Z. and Yin, M. (2023). Synthetic data generation with large language models for text classification: Potential and limitations.
URL: *https://arxiv.org/abs/2310.07849*

Liao, M., Huang, L., Zhang, J., Song, L. and Li, B. (2024). Enhanced hs code classification for import and export goods via multiscale attention and ernie-bilstm, *Applied Sciences* **14**(22).
URL: *https://www.mdpi.com/2076-3417/14/22/10267*

Liu, N. F. et al. (2024). Lost in the middle: How language models use long contexts, *Transactions of the Association for Computational Linguistics* **12**: 157–173.

LL.M., N. K. and Romaní, M. S. (2022). *AI biases and its consequences on taxation*, 1 edn, Maastricht University Press. https://pubpub.maastrichtuniversitypress.nl/pub/ai-biases-and-its-consequences-on-taxation.

Perçin, S. et al. (2025). Investigating the robustness of retrieval-augmented generation at the query level, *in* O. Arviv et al. (eds), *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM)*, Association for Computational Linguistics, Vienna, Austria, pp. 439–457.
URL: *https://aclanthology.org/2025.gem-1.38/*

Shubham et al. (2023). An ensemble-based approach for assigning text to correct harmonized system code, *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pp. 35–41.

Siino, M., Tinnirello, I. and La Cascia, M. (2024). Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers, *Information Systems* **121**.

Siva, P. K. (2025). Legal and regulatory implications of agentic ai in tax administration, *Tax Notes International* **118**. Available at SSRN: https://ssrn.com/abstract=5333340.

Yuan, Y., Li, X. and Yang, Y. T. (2020). Punctuation and parallel corpus-based word embedding model for low-resource languages, *Information* **11**(1).