An Ensemble-based approach for assigning text to correct Harmonized system code

Shubham
Consulting
IBM India Pvt Ltd
Bengaluru, India
shubham1@ibm.com

Avinash Arya
Consulting
IBM India Pvt Ltd
Noida, India
avinarya@in.ibm.com

Subarna Roy
Consulting
IBM India Pvt Ltd
Bengaluru, India
subroy12@in.ibm.com

Sridhar Jonnala
Consulting
IBM India Pvt Ltd
Bengaluru, India
srjonnal@in.ibm.com

Abstract—The world customs organization (WCO) administers the Harmonized System (HS) that provides guidance on common nomenclature to describe traded goods by names and numbers. Inaccurate or incomplete usage of HS codes lead to regulatory violations by the company for which the goods are traded. Shipment often gets delayed because of the inaccuracies encountered. The customs often classify these products into "other" category which turns out to be expensive as high tariff rates are charged. Hence a system is required for accurate assignment of text-description to HS code as per WCO manual. In order to develop an accurate commodity classification engine this paper has provided solutions under two scenarios. Under the first scenario where sufficient and reliable training data is available, we build a BERT-transformer based hierarchical conditional multi-class classification model that outperforms the flat classification model by approximately 16 percentage on a publicly available data. As the HS codes are hierarchical in nature, building a model at each digit gives a clear edge. The reason is that the algorithm arrives at the final class by traversing all the possibilities at each digit level conditional upon the digit chosen at previous level. Under the second scenario, where training data is unreliable as is often the case due to unverified user input stored in the backend, we demonstrate how correct assignment can be arrived at by modelling the WCO manual as a NER based knowledge-graph and then computing the similarity of example text description with the nodes and edges of the graph. We assign the text-description to a HS code based on the highest similarity measure. The knowledge-graph also introduces AI based auditing functionality in the system which is a novelty and addresses questions of responsible AI.

Keywords— HS code classification, Bert-Transformer, Knowledge-graph, AI based auditing

I. INTRODUCTION

World Customs Organizations administers HS code globally to classify imports and exports related to international trade. These codes inform the customs officials about the kind of goods being imported and exported. They help in assessing the duties, taxes, and other fees.

These codes are comprised of a hierarchically structured nomenclature which enables industries in US and its trading partner countries to classify their products till the 6-digit level following the HS manual [1] published by world customs organization (WCO). The HS manual is updated every 5 years.

The trading entities also need to share the methodology that has been used to come up with the assigned HS code. The methodology should adhere to the General rules of Interpretation (GRI) norms. GRI comprises of set of 6 rules that are used to identify products. These rules need to be administered in a sequential manner to ensure uniform legal understanding of the HS codes.[2]

There has been exponential growth in traded volumes in recent years, custom officials are facing major challenges to ensure accurate collection of taxes and duties. Data analytics and AI have shown promising interest to address current challenges [3] [4]. HS code is widely accepted by customs officials for taxes and duties declarations. Abdolshah et al. [5] explored computer vision techniques to verify declarations made on shipment containers. Turhan et al. [6] used textual analysis on product description to assign HS code automatically. Beyond 6-digit classifications, countries are allowed to assign their own custom HS codes at 8–10-digit level. Taking an example from [7], the hierarchical structure of the code is as follows:

![Result](https://via.placeholder.com/150)

Fig. 1. HS code classification hierarchy

However, based on information from various third-party vendors that collects such data, it has been found that only 29% of the text that are entered by the users for shipments and computation of tariffs can be automated and assigned to the right HS code albeit with certain accuracy. This is because firstly, users are allowed to enter free text which generates messy /noisy/ non-sensical data at the backend. Secondly, to avoid higher tariffs or to take advantage of exemption policies users often resort to entering ambiguous texts, making code assignment challenging for the system. Noisy and ambiguous texts require manual intervention which is often error-prone leading to high degree of misclassification and revenue leakage. Over the counter deep-learning text classification models or machine-translation models may not be of help in automating such texts. Based on data available in the public domain we have broadly classified such texts as follows:
Also, a larger view of the document-hierarchy demonstrates the complexity of the classification problem. As one goes to the bottom of the hierarchy the degree of cardinality increases- rendering flat multi-label text-classification models unstable. They lack the capability to generalize as well.

While, multi-label classification models are useful at 1st two levels, at more granular levels the following challenges are observed:

A classification model at XX-digit level (which provides unconditional probabilities for a text description to belong to a particular class at XX-digit) might require thinly populated classes to be grouped into ‘other’ category. Thus, new text description that has a potential to fall into this category will miss out on being labelled into a specific class- leading to manual intervention. If they are not grouped, then the model will lack generalization. For example, let’s say 76.01.20 is a thin-volume class and hence has been grouped with rest of the thin volume classes forming a ‘other’ category during training. If a new text-description has a potential to belong to this class e.g., 76.01.20, no way from the model it can be determined that it would be the case.

Secondly, a flat classification model would miss out on exploring the possibilities of belonging to a specific class or set of classes at each level. Flat model lacks the capability to traverse the search-space of each level and use that information in classification at the final level- thus eroding models’ ability to capture nuances. Hence, model training should be hierarchical in nature. The theoretical framework for this kind of model training will be explained in section-III.

Also, since problem related to thin volume is more severe at lower end of the classification pyramid, supervised learning by leveraging companies own data should be restricted till the 4-digit level even when the hierarchical option of model training is exercised. At 6-digit level and beyond using a combination of unsupervised learning methods e.g., sentence-embedding, named entity recognition [NER], semantic search [8], cosine-similarity and knowledge graph can yield better result. Therefore, coverage of the system to classify more text-description in specific code increases.

We demonstrate the superiority of the later approach using publicly available dataset [9] based on several metrics – e.g., scalability, coverage, generalization, ability to capture nuances and variations in text etc. We further show how knowledge-graph can be designed based on the WCO manual. The proposed methodology is a useful utility to highlight missing information in user-provided text. This serves the audit requirement of providing rationale for HS classification.

II. METHODOLOGY

A. Related Work

Various approaches and techniques have been adapted for the task of HS classification, ranging from SVMs to Neural network models. Ding et. al [10] utilized a Background Net (B-net) approach for the automatic classification of HS codes. HS code classification can be seen as a hierarchical classification problem due to its hierarchically structured nomenclature. Flat approaches model the hierarchical classification problem as a multi-class classification problem. A wide range of flat methods applied to this problem of classification include Support Vector Machines [11], and Neural networks [12] among others. There are hierarchical local approaches which include local classifiers for different levels and are isolated from each other [13, 14].

B. The Flat Model

The outcome of training a flat model is the unconditional probability for a given feature vector $x_i$ extracted from $i$th description to belong to any one of the $K$ classes. Each class comprises of one 6-digit HS code. Let’s denote the predicted class by $\hat{y}_i$, then the unconditional probability of it being equal to the $j$th class is given by the SoftMax function:

$$p(\hat{y}_i = j) = \frac{e^{x_{ij}}}{\sum_{k=1}^{K} e^{x_{ik}}}, \forall j = 1 ... K$$  \hspace{1cm} (1)
In this approach a BERT [15] – based sequence classifier is used. The BERT Encoder layer is kept trainable, which allows us to fine-tune it according to the data and improve the model performance. The output layer of the BERT model is replaced by the soft-max layer as per (1) to meet the requirement of multi-label classification. A typical architecture is represented below:

![Architecture Diagram](image)

**C. The Hierarchical Model**

This modelling approach is a 3-step process. First, a 2-digit classifier is built in the same way as explained in subsection B for a 6-digit classifier. Let us say there are \( K^{(2)} \) classes at 2-digit level. Hence, the unconditional probability of predicted 2-digit class \( y_i^{(2)} \) being equal to \( l \) is given by:

\[
p(y_i^{(2)} = l) = \frac{e^{x_{l,d}}}{\sum_{k=1}^{K^{(2)}} e^{x_{k,d}}} \quad \forall l = 1 \ldots K^{(2)}
\]  

(2)

The same architecture as explained in Fig.4 is followed to obtain a 2-digit classifier for a set of text-descriptions. The 2-digit class with the highest probability of occurrence for the ith observation is obtained based on minimizing the cross-entropy loss function in (3)

\[
C = \sum_l \sum_{k=1}^{K^{(2)}} - \hat{y}_i^{(2)} \log p(y_i^{(2)} = l)
\]

(3)

Let, \( l^* \) be the class with the highest probability of occurrence for ith observation

\[
p(y_i^{(4)} = l^*) = \max_k \sum_{k=1}^{K^{(2)}} p(y_i^{(2)} = l^*)
\]

In the next step, the probability of predicted 4-digit class \( y_i^{(4)} \) being equal to \( m \) can be computed as,

\[
p(y_i^{(4)} = m) = \frac{e^{x_{m,d}}}{\sum_{k=1}^{K^{(4)}} e^{x_{k,d}}} \quad \forall m = 1 \ldots K^{(4)}
\]

(4)

This is under the assumption that \( K^{(4)} \) is the number of 4-digit classes under \( l \)th 2-digit class. Hence, the conditional probability that predicted 4-digit class \( y_i^{(4)} \) being equal to \( m \) given predicted 2-digit class \( y_i^{(2)} \) is equal to \( l^* \) can be written as:

\[
p(y_i^{(4)} = m | y_i^{(2)} = l^*) = \frac{e^{x_{m,d}}}{p(y_i^{(2)} = l^*)} \sum_{k=1}^{K^{(4)}} e^{x_{k,d}}
\]

(5)

Please note that in our current empirical exercise in section - III, the unconditional soft-layer at 4-digit level is applied based on (4). Also, for current implementation the training dataset for this level contains all observations that have historically been assigned a class \( l^* \) and not only those for which \( l^* \) is the class with highest probability of occurrence.

The 4-digit classifier is computed using (4), corresponding cross entropy function and the architecture in Fig.4.

At the third step, distance-based unsupervised modelling approaches are considered by comparing the organization’s dataset at 6-digit level first with training set and then with the WCO manual. If the \( i^{th} \) text-description falls in 4-digit class \( m^* \) based on maximum probability of occurrence, then it is compared will all the 6-digit classes under \( m^* \) in the training dataset using cosine similarity. BERT-based sentence embedder is leveraged to convert all the text-descriptions into vectors including the \( i^{th} \) text-description for which comparison is made. The cosine similarity between \( i^{th} \) text-description and \( o^{th} \) text-description from the training set is computed based on the following equation:

\[
\rho(i, o) = \sum_j \cosine(i, o)
\]

(6)

Let’s say, \( o^* \) is the class for which \( \rho(i, o) \) is maximum. So \( o^* \) is chosen as the 6-digit class for \( i^{th} \) text-description.

The above approaches all rely on the training dataset to male predictions. But as demonstrated in fig. 2 the historical dataset is highly susceptible to human-errors. Users have to fill-up a form while sending their shipment. The data created is the digital copy of what users enter manually in the form. Given the extent of data created every day, practically it is not possible to validate each row. So, over-reliance on training dataset to predict HS code in future would aggravate the problem of incorrect assignment rather than solving it.

Secondly, there is frequent changes in the WCO manual which makes the rules that were used for previous assignments invalid. Hence, over-reliance on training data under this changing circumstance will render HS code prediction almost ineffective. Therefore, at the third step of the methodology we also suggest comparison of the text with WCO manual to get alternative suggestions for HS code which might be more appropriate than what is assigned based on past examples.
The end-to-end methodology of comparison with WCO is described in the next subsections. In this case, all the 6-digit classes under \( m^* \) in WCO is modelled as a knowledge-graph through custom rule extraction. Let’s say, \( m^* \) corresponds to the 4-digit class 84.14. Fig.5 demonstrates some of the 6-digits and 8-digits codes and standardized description under this class. The process-flow is as follows:

- Extraction of customs-rules from WCO descriptions
- Visualization the rules in the form of a knowledge-graph
- Comparison of entities in the 6-digit descriptions with nodes and links in the knowledge graph and compute average cosine-similarity
- Select the 6-digit class that corresponds to the maximum average cosine-similarity

Fig. 5. Descriptions for headings and sub-headings of 8414. The digits highlighted in red are either 6-digit classes or classes which have no further sub-headings post 6-digit

D. Custom-rule extraction

To extract entities and the relationships between them, we created a rule-based entity-relationship extraction algorithm which given a sentence splits it into entities and relationships. The sentence is first POS tagged and passed to the rule-based extractor which works upon the POS tags. Through POS tagging, a dependency graph is generated which can be traversed using certain rules to identify entities and relationships.

E. Knowledge-graph

Post segmentation of the text corresponding to a given 6-digit into meaningful phrases, the entities containing nouns/pronouns are represented as nodes. The contextual word to the nouns also goes to the relevant nodes. The rest of the entities are considered as linking phrases and assigned to the edges.

F. Entity comparison with distance measures

The \( i^\text{th} \) text that has a maximum probability of occurrence in the 4-digit class of \( m^* \) is then compared with each node and links of the knowledge graph of all the associated 6-digit classes. Let’s say there are \( K^m(6) \) 6-digit classes under \( m^* \) 4-digit class. If there are \( N \) nodes and links within the \( k^\text{th} \) 6-digit class in accordance with the construction of the knowledge graph, then the average cosine similarity \( (\rho^o) \) is computed based on the following:

\[
\rho^o(i, o) = \frac{\sum_{j=1}^{S} \cosine(o^{(j)}, o^{(i)})}{S^o} \tag{6}
\]

\( o^{(j)} \) is the \( j^\text{th} \) node/link of the \( o^\text{th} \) 6-digit class that belongs to \( m^* \) 4-digit class. Let’s consider \( o^* \) is the class for which \( \rho^o \) is maximum. Then the \( i^\text{th} \) observation will be assigned the \( o^* \) 6-digit class. For the task of execution of cosine similarity methods, first the sentences/phrases from product-description and HS code descriptions in WCO are vectorized using sentence-BERT base version.
In the implementation, we considered top 3 most probable HS codes in terms of average cosine similarity instead of only the top-most. The indicative high-level workflow is given below:

![Fig. 8. Comparison of text based on cosine-similarity and knowledge-graph approach](image)

This approach is like [16], but one significant difference is the custom rule extraction layer that extracts entities and links from the HS code description. Each entity and link are then used to form the knowledge graph. The pre-processed product description is then compared with each of these entities/links to come up with the average cosine-similarity score. Fig. 7 illustrates the approach where a pre-processed product description is compared with three knowledge-graphs, the nodes, and links of each of which is extracted from HS code description in WCO manual. One that has the highest match in terms of average cosine-similarity is chosen. Post comparison the color of each graph changes to green, light-green, and yellow depending on the extent of the match with the highest match represented by green. In case, there is no match the node or link color remains blue.

### III. EMPIRICAL APPROACH

#### A. Data

The dataset used for HS code classification is a publicly available data. This dataset contains cargo shipment description, harmonized code, and other features. Product descriptions having been assigned with 6+ digit HS code have been used for training Flat 6-digit model and Hierarchical model. Exogenous variables- harmonized weight and harmonized value have been used in model training. HS4 Classification model is trained on Chapter 84 since it is the class with the highest volume among all the 4-digit classes. Table I contains key details about train data and test data for flat and hierarchical model.

As is evident from Table I, very few classes have significant volume (> 5%), more so for HS6 classes, though the number of classes in this category is the highest, therefore many classes within it have negligible volume rendering direct classification ineffective.

#### B. Pre-processing steps

Before using the data for modelling, we apply some pre-processing steps which yields in better model performance. The pre-processing steps vary according to the data. Some datasets have very concise and accurate descriptions, while some have descriptions which require a lot of cleaning and processing.

1. **Text pre-processing steps:**
   a. Remove extra-spaces, special characters which do not provide meaning to the description
   b. Remove Email IDs, phone numbers, fax numbers and other unnecessary texts
   c. As these are manually written descriptions, there might be incorrect spellings, we correct such words using various techniques to a certain extent of error.
   d. Descriptions also contain certain abbreviations which can be specific to that company alone. We replace certain known abbreviations with the actual word.
   e. Lemmatization

2. **Ambiguous description data points:**
   Certain descriptions in the dataset are assigned to multiple HS codes, which can lead to decreased model performance. Therefore, in a particular ambiguous description, if a HS code occurs more than 80% of the times, we retain that description-HS code combination and remove the rest.

3. **Low frequency classes are grouped together to form a ‘others’ class.

4. **Standardizing numerical features**

### IV. RESULTS

Post pre-processing, both flat and hierarchical models were trained as per the methodology described in section 2. The accuracy was computed with test sample of 300 observations. The results are provided below. There is a 16% increase in accuracy if hierarchical model is implemented vis-à-vis a flat model. The flat model also was unable to classify 12% of the text-descriptions into legitimate classes. While the proposed hierarchical model assigned valid classes to 100% of the text-descriptions. Both the models were trained using transformers and demonstrated outstanding ability to capture nuances. For example, the flat model was able to rightly classify the text-description “casting parts finned aluminium air cleaner hb” in ‘76’, though it was originally classified in ‘84’. Similarly, “mens polo style pm1001 style” was rightly classified as ‘62’, though it was originally assigned to ‘84’. One more interesting result is the assignment of “pallet sodium trifluoroacetate 25 0kk net weight 100 0000 kgs ams hbl 1855728061 scac bopt” in ‘39’ which is a correct assignment for chemicals, while its original classification was ‘84’.

| Model                  | Train Data Size | Test Data Size | No. of classes | No of classes with volume greater than 5% |
|------------------------|-----------------|----------------|----------------|------------------------------------------|
| Flat (HS6) Model       | 232467          | 77490          | 555            | 5                                        |
| Hierarchical (HS2) Model | 232467       | 77490          | 54             | 5                                        |
| Hierarchical (HS4) Model – Chapter 84 | 54471         | 18158          | 44             | 6                                        |

39
We also demonstrate the superior classification ability when texts are compared with knowledge-graph extracts of WCO manual for distance calculation. For example, “package stc conical roller bearings” has been originally assigned a HS code of 848291 which as per the WCO manual is referred to texts related to “Balls, needles and rollers: “. However, when all related knowledge graphs are extracted based on methodology described in section II, the knowledge-graphs related to the HS codes (848250, 848251) appeared to be more relevant in terms of similarity measure.

In Fig. 9, since the node of 848250 has the keyword ‘cone’ in it, while 848251 has the keyword ‘cylindrical’ along with the keyword roller bearing, understandably, when the cosine similarity is computed these two codes got higher score than 848291. The code 848291 has only the keyword ‘roller’ in the node.

Results from knowledge-graph should be correlated with the GRI rules to come up with the final recommendations. For example, GRI Rule 3 mentions that "When by application of Rule 2 or for any other reason, goods are, prima facie, classifiable under two or more headings, classification shall be executed as the heading which provides the most specific description shall be the preferred headings choice compared to a more generic description. Unless it refers to only part of the mixed substances or part of the items put up for retail sale.” [2]

| Model-No. | TABLE II: Accuracy and coverage by model type | Accuracy | Coverage |
|-----------|--------------------------------------------|----------|----------|
| 1         | Flat classification model(at 6 digit level) | 56%      | 88%      |
| 2         | Hierarchical classification(2+4+6 digit)    | 65%      | 100%     |

Hence, the proposed methodology by its inherent construction adheres to the GRI norms.

We have discussed on approaches that would provide superior results when training dataset is not reliable or there is continuous update in HS classification codes at a global level and sometimes at individual country level.

V. CONCLUSIONS AND FUTURE SCOPE

We have discussed and compared the two different approaches – the hierarchical and the flat approach to address the HS code classification problem. While the hierarchical methodology with its conditional probability approach ensures that texts get assigned to right code with a greater accuracy, imbalance in data translates greater biasedness towards low-level classification (HS4, HS6)- eventually impacting the hierarchical model as well. Such high imbalance is not visible for Flat Model. Another important contribution of this paper is the interpretability through knowledge-graph. We have demonstrated the ability of the solution to show why a particular text is being assigned to a specific HS code- which would help in audit purpose. This approach introduces explain-ability in the AI solutions and has the visual functionality to inform the custom officials about the process through which the most appropriate HS code is arrived at. Hence, the knowledge graph approach goes a long way in addressing the questions related to responsible AI. With this also we eventually expect to build a system that will encourage users to follow the GRI norms while classify products into HS codes. This will result in better collection of data and thereby improving the decision-making system. No
other work has implemented such a solution to our knowledge.

Our Future work in product HS code classification is focused on the following:

- Enhance sentence embedding model using industry specific knowledge to fine-tune current embeddings for increased prediction accuracy
- Address biasedness problem by building the overarching model of the hierarchical methodology at industry level rather than at a chapter level
- Penalizing model when wrong prediction using reinforcement learning

ACKNOWLEDGMENT

We would like to express our sincere thanks to our colleague in IBM Consulting – Avinash Kalyanaraman for his valuable inputs

REFERENCES

[1] World Customs Organization. HS compendium – The harmonized system, a universal language for international trade. http://www.wcoomd.org/-media/wco/public/global/pdf/topics/nomenclature/activities-and-programmes/30-years-hs-hs-compendium.pdf.

[2] What is GRI and How It Can Affect Your HS Classification https://globalextrade.services/blogs/what-gri-and-how-it-can-affect-your-hs-classification

[3] F. Heijmann, Y. Tan, B. Rukanova, and A. Veenstra, “The changing role of customs: Customs aligning with supply chain and information management,” World Customs Journal, vol. 14, no. 2, 2020.

[4] B. Rukanova, Y.-H. Tan, M. Slegt, M. Molenhuis, B. van Rijnsoever, J. Migeotte, M. L. Labare, K. Plecko, B. Caglayan, G. Shorten et al., “Identifying the value of data analytics in the context of government supervision: Insights from the customs domain,” Government Information Quarterly, vol. 38, no. 1, p. 101496, 2021.

[5] M. Abdolshah, M. Teimouri, and R. Rahmani, “Classification of x-ray images of shipping containers,” Expert Systems with Applications, vol. 77, pp. 57–65, 2017.

[6] B. Turhan, G. B. Akar, C. Turhan, and C. Yukse, “Visual and textual feature fusion for automatic customs tariff classification,” in 2015 IEEE International Conference on Information Reuse and Integration. IEEE, 2015, pp. 76–81.

[7] Xi Chen, Stefano Bromuri, Marko Van Ekelen, Neural Machine Translation for Harmonized System Codes prediction, https://dl.acm.org/doi/fullHtml/10.1145/3468891.3468915

[8] Yinfei Yang and Chris Tar. Advances in semantic textual similarity, 2021. https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html

[9] US Imports CBP Automated Manifest System (AMS) Shipments 2020, https://aws.amazon.com/marketplace/pp/prodview-stk4wn3nbhx24#datasets

[10] Liya Ding, ZhenZhen Fan, and DongLiang Chen. Auto-categorization of hs code using background net approach. Procedia Computer Science, 60:1462–1471, 2015

[11] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. N’edellec and C. Rousseil, editors, Proceedings of the 10th European Conference on Machine Learning (ECML’98), number 1398, pages 137–142. Springer Verlag, 1998.

[12] A. S. Weigend, E. D. Wiener, and J. O. Pedersen. Exploiting hierarchy in text categorization. Information Retrieval, 1(3):193–216, 1999.

[13] L. Cai, T. Hofmann, Hierarchical document categorization with support vector machines, in: Proceedings of the thirteenth ACM international conference on Information and knowledge management, 2004, pp. 78–87

[14] N. Cesa-Bianchi, C. Gentile, L. Zaniboni, Hierarchical classification: combining bayes with svm, in: Proceedings of the 23rd international conference 280 on Machine learning, 2006, pp. 177–184

[15] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint, 2018.

[16] Kaustav Pain, “Harmonized System Code Classification Using Transfer Learning with Pre-Trained Weights”, https://dalspace.library.dal.ca/handle/10222/80672