HARMONIZED SYSTEM CODE CLASSIFICATION USING TRANSFER LEARNING WITH PRE-TRAINED WEIGHTS

by

Koustav Pain (Tukai)

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
July 2021

This thesis is dedicated to my loved ones, who always believed in my abilities. You are gone but your belief in me has made this journey possible.

Contents

List of Tables

List of Figures

List of Abbreviations Used

ADD	Anti-Dumping Duties
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
CIMT	Canadian International Merchandise Trade
CN	Combined Nomenclature
CVD	Countervailing Duties
DL	Deep Learning
EDI	Electronic Data Interchange
EU	The European Union
GLUE	General Language Understanding Evaluation Benchmark
HS	Harmonized Commodity Description and Coding System
IP	Intellectual Property
ITC	Indian Trade Clarification
ML	Machine Learning
NER	Named Entity Recognition
NLI	Natural Language Inference
NLP	Natural Language Processing
NRC	National Research Council Canada
NSP	Next Sentence Prediction Task

$RFID$	Radio-frequency Identification
STS	Semantic Textual Similarity
SVM	Support Vector Machine
$TARIC$	TARif Intégré Communautaire; Integrated Tariff of the European Communities
$TF-IDF$	Term Frequency-Inverse Document Frequency
TL	Transfer Learning
UN	The United Nations
USE	Universal Sentence Encoder
WCO	World Customs Organization

Abstract

The Harmonized System (HS) was developed by the World Customs Organization (WCO) as a multipurpose international product nomenclature that describes the type of good that is shipped. Nearly universal usage of HS allows customs authorities to identify and clear every commodity that enters or crosses any international borders. HS code classification is the task of identifying the HS code of a commodity according to its description information provided in a trade document. In fact, the HS code classification is fundamentally a text classification problem. However, compared with general text classification, the challenge of this task is that the commodity description texts are often very short, unstructured, and extremely noisy. There are more than six thousands HS codes for different commodities. What's more, the label space descriptions for a commodity; i.e., the HS nomenclature, often differs from how shippers like to describe commodities in a trade document. HS misclassifications or using an incorrect HS code can lead to penalties, fines, and delays upon import.

In this research work, we first propose novel approaches for extracting and filtering relevant commodity information from a trade document. Next, we propose an HS code classification methodology that utilizes existing pre-trained Semantic Textual Similarity (STS) models via deep transfer learning using sentence-level transfer. We also introduce a new evaluation method to properly evaluate our approach based on real-world applications. Extensive experiments and model comparisons show the superiority of our approach.

Acknowledgements

I want to thank my supervisor, **Dr. Vlado Keselj**, without whom none of this would have been possible. Dr. Keselj has supported me throughout the process and has been a constant source of encouragement. The first course I ever took at Dal was Advanced topics in NLP taught by Dr. Keselj. The highly dense and structured course sowed the seed of growth for my theoretical and technical knowledge in NLP. Dr. Keselj has helped me sharpen my skills at identifying, contributing to, and producing research that is aligned with my interest. Due to his valuable lessons and feed backs, I was able to learn and navigate through Latex from no experience to a great detail.

I want to thank our Dal **DNLP group** consisting of extraordinary individuals whose outstanding work and feature presentations inspire me every day.

I am thankful to **Dr. Kirstie Hawkey**, **Dr. Stan Matwin** and **Dr. Stephen Brooks** whose outstanding courses helped me gain knowledge in the domain of research methods and statistics, Deep Learning, and the importance of Visualization. The knowledge gain kept me aware of fundamental aspects of research such as internal, external, and statistical validity of the research, explainable AI, and technical scalability of the research.

I want to thank my parents, **Dr. Kalyan Prasad Pain** and **Chitra Pain**, my sister, **Koushiki Pain** and my partner, **Marsha Wangsadiredja** for everything. Thank you for letting me travel to study on the other side of the world. I want to thank all my friends, who despite a pandemic never forgot to keep checking on me. I miss you every day.

Finally, I'd like to thank the DeepSense team especially **Jennifer LaPlante** and **Jason Newport**, and Mitacs Canada for this research opportunity with the partner company.

Chapter 1

Introduction

The World Bank notes Canada exported approximately $450 billion worth of goods in 2019, and imported $459 billion worth of goods. This represents approximately $1 trillion of international trade which represents over 4,300 products exported to over 200 countries and over 4,500 different products imported from almost 225 countries (based on the HS6 commodity codes) [9]. The Harmonized Item Description and Coding System (HS) maintained by the World Customs Organization (WCO) is a standardized numerical method of classifying traded products. It is used by customs authorities around the world to identify products when assessing duties and taxes and for gathering statistics. The HS assigns specific six-digit codes for varying classifications and commodities. Countries are allowed to add longer codes to the first six digits for further classification [10]. Both export and import statistics used in the Canadian International Merchandise Trade (CIMT) database are available at the HS-06 detailed level [11].

In Canada, the traded goods arrive and depart through a variety of trade channels, most principally rail, maritime shipping, truck cargo and air freight. The data reflecting the full scope of this trade, across all the various channels is transmitted and stored in EDI files. EDI, which stands for Electronic Data Interchange, is the inter-company communication of business documents in a standard format. The simple definition of EDI as defined by IBM, is a standard electronic format that replaces paper-based documents such as purchase orders or invoices [12]. In EDI transactions, information moves directly from a computer application in one organization to a computer application in another. By automating paper-based transactions, organizations can save time and eliminate costly errors caused by manual processing.

However, the data associated with each shipment is not always accurate and reliable. The details about each shipment are critical as the duty and taxes charged for the import of the goods in Canada via cargo terminals such as air freight, maritime shipping, rail or truck are vital to the effective control of commodities. There is a significant need to rectify the HS commodity codes to ensure it aligns with the appropriate commodity. Since the information is typed in an EDI shipment file, there are often errors like missing fields, wrong HS commodity codes, codes present in wrong fields which are nearly impossible to distinguish from other digits in a trade document such as fax numbers, phone numbers, postal codes etc. Thus, the processing of cargo today requires significant manual effort to review, correct and rectify errors. Since EDI was designed for use by primitive machines in the 1960s, some older formats of EDI (EDIFACT or IMP) were not meant for humans to read and understand them, therefore catching and fixing errors can be challenging [13]. Even when it is possible to read the information, manual error detection is time-consuming and expensive, and can still result in errors. We believe that automating error detection and fixing them can significantly increase productivity and increased gross margins.

Each EDI shipment file should include commodity codes and some manually entered text to identify the contents. While some data is accurate, there remains a portion of data that is entered in the wrong field or is entirely inconsistent. Shipments may contain a single item, multiples of one item or be filled with a broad range of unrelated items (i.e., a family relocating to a new country would include a range of commodities). A primary ambition of this research is to automate processes of shipment validation and reduce risk. A lot of the operations are currently performed by manual labor, including shipping container code assignment. Transactions are processed by thousands every day, and the process is prone to human error — an example would be, an EDI trade document with an apple in the commodity description and a commodity code that is of a bicycle.

HS code classification can be regarded as a text classification task. However, compared to general text classification, it faces the following challenges:

1. **Complexity of nomenclature.** The HS is a structured multipurpose nomenclature, organized into 21 sections and 97 chapters. The World Customs Organization (WCO) has developed a large number of defined regulations, such as notes, subheading notes, and code structure explanations, to aid customs officers and other professionals, but not traders. These contribute to the difficulties associated with properly classifying products in the HS [14]. The order

of arrangement of commodities in terms of HS is based on their degree of manufacture. Within each chapter, the headings are arranged based on a principle.

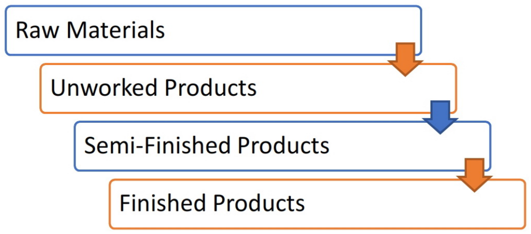
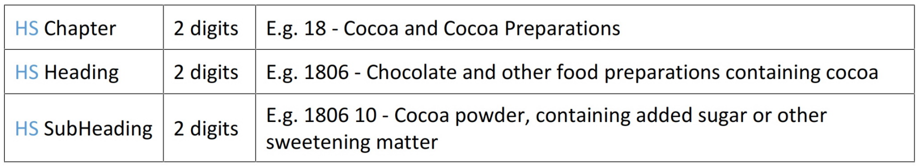


Figure 1.1: Principle of order in HS [1]

Chapters are broad level categories that divide commodities into categories such as coffee, cocoa, tea, and wine. The product's additional specs are clarified by the headings and subheadings.



HS Chapter	2 digits	E.g. 18 - Cocoa and Cocoa Preparations
HS Heading	2 digits	E.g. 1806 - Chocolate and other food preparations containing cocoa
HS SubHeading	2 digits	E.g. 1806 10 - Cocoa powder, containing added sugar or other sweetening matter

Figure 1.2: Each section indicates further complexity in the classification [1]

2. **Terminological gaps.** There is consistently a major gap between merchandise portrayal of a traded commodity and their respective depiction in UN-defined HS terminology. For instance, shippers might want to define a product as an "MP3 player", however, they need to understand that it belongs to "85.19 — Sound recording or reproducing apparatus: Other apparatus: 8519.81 Using magnetic, optical, or semiconductor media" [15]. Simple string search is unable to help either customs or the shippers to locate the relevant HS codes because of the differences between the structured descriptions of HS nomenclature and the text descriptions during the trade process.

3. **Differences by nation.** Many countries append further classification digits to the right of HS codes to add their own classification levels. For example, HTS code is the US-specific extension of HS codes. EU uses 8-digit CN (Combined Nomenclature) and 10-digit TARIC (TARif Intégré Communautaire; Integrated Tariff of the European Communities) codes [16, 17]. China uses 13-digit HS

codes and India uses a standard called ITC-HS codes (Indian Trade Clarification). In most cases, these HS codes need to be converted to the country specific standard during import.

4. **Ever-changing nature.** A revision of 6-digit HS codes occurs every five years [15]. National HS codes are also updated more often, up to a few times each year in some cases. This necessitates the use of a classification system that is both reliable and adaptable to changing commodity descriptions.

To address the above challenges, we first present a novel HS code parser using natural language processing concepts to extract the codes and associated text including variances in language. Second, we present four different shipper-specific novel data clean-up methods to clean up the text descriptions of EDI trade documents reducing it down to only the description of the commodity being traded. Third, via GPU acceleration we present a HS code recommendation system that can process ten thousand trade transactions in under a minute. Our system leverages the pre-existing concept of deep transfer learning for natural language processing via pre-trained sentence transformers to generate and compare sentence level embedding to determine if the commodity codes entered in a trade document should be edited.

1.1 Motivation

One can think of HS code classification as the last challenge goods face before hitting the finish line in the shipping process. These 6-10 digit numbered harmonized tariff codes serve two major roles and purposes aside from helping goods clear through customs:

1. They identify products that are being imported or exported through a country's borders.

2. They classify and categorize products in a worldwide system used for customs clearance purposes.

Before getting into the importing and exporting market, the most important step for shippers to take is assigning an HS Code. Importers and exporters might underestimate the importance of the HS code. Most commonly, suppliers just accept an importer's purchase orders and ship their goods without knowing a good's HS code. Shipping before assigning an HS code is not a good business practice and can cause serious problems that could be costly, harmful to both parties' core businesses, and damaging to one's reputation [18]. Unless you want to have your goods seized or lose your import privileges it is recommended shippers research the proper commodity code. Correct commodity code classification is an important piece of information, especially when using an incorrect classification can lead to penalties and transit delays.

The risks can be viewed from every side; i.e., a shipper, receiver (ports, airports, cargo rail station etc.) and the government.

From the shipper's side, failure to classify correctly can lead to: non-compliance penalties, border delays, seizure of goods, fines, denial of import privileges, and other consequences.

Since there are multiple stops between cargo transports, from the receiver's perspective failure to correctly classify HS codes can lead to:

1. Fires breaking out on container ships due to improperly declared dangerous goods [19].

2. Fraudulent shipping codes being used to avoid shipping restrictions or custom duties. Detecting fraud is often done through random audits, however this is labour intensive due to the massive amount of cargo shipped each day [20].

3. If handled carelessly, the possibility of transmission or even outbreak of the recent Covid-19 Virus due to its long survival capabilities depending on the surface material of the goods [21].

4. Chemical contaminants causing hazards, infectious diseases.

For government, correct classification is required for three main purposes [15]:

1. Calculation of duties, taxes and fees. 2. Determination of permits, license and certificates required. 3. Collection of trade statistics.

So, we see the importance of HS code classification being very significant. Providing the correct HS code is a part of both the importer's and exporter's legal responsibilities. It is recommended that both parties need to make sure that they have included all the necessary classification resources and that they take time to get them right — no matter the assortment of products being imported or exported [18].

1.2 Contributions

The main contributions of this thesis document can be summarized as follows:

1. We have a deep insight into the HS commodity code classification problem. We explored its importance, uniqueness and challenges compared with the typically studied text classification problems. We also explored three different means of text classification to accomplish our goals.

2. We proposed a novel HS code parser able to extract the HS commodity codes from extremely tangled and complex text descriptions.

3. We proposed and evaluated novel commodity description clean-up approaches based on real-world dataset of several transportation companies to clean the description part of the transactions in an EDI trade document.

4. Our proposed approach for HS commodity code classification and evaluation makes it an appealing method for extremely noisy and unstructured trade data. Due to the sensitivity of the data, our approach also follows a safer way that attempts to avoid HS code misclassifications, resulting in sparing of multiple long occurring and sometimes severe consequences (discussed in section 1.1).

5. The usage of the state-of-the-art GPU acceleration via NVIDIA CUDA and TensorFlow-GPU ensures the scalability of our approach. Our evaluation methods have a strong focus on interpretability and explainable AI. The proposed approach has shown positive results and it has been adopted by the client authorities in their production system.

1.3 Research Overview

During the initial stage of the research work, one major obstacle in working with the data consisting of EDI files is that there are many organizations using it and they use different styles and different ways of writing product information. There is no general guidebook for this information and the only way to derive information clean-up procedures is to go manually through the samples of data. Deriving the clean-up procedures also required exploring different internet sources such as customs websites from different countries such as Canada, US, India, and Spain.

An in-depth literature survey was conducted to find out if there are any existing systems that can even partially address the problem at hand. The focus at this stage was to find any relevant existing work and look for improvements. A hope to be able to continue the research from there. There are more relevant information available in online web articles than academic research papers. From what it seemed, the research topic at hand was highly underrated so far. Most of the work was being conducted with manual human effort disregarding the errors.

1.4 Roadmap of the Thesis

This section provides a roadmap that describes the content of each chapter in the thesis.

Chapter 2 contains the background definitions and literature review. In this chapter, first, the importance of commodity code is explained. After that, different types of terminology related to global trade and HS coding system are mentioned. Finally, we discuss the knowledge gap our research is trying to fill.

Chapter 3 narrows down the research problem and also represents the thesis contribution. This chapter also describes different data sources and datasets used to conduct the research.

Chapter 4 consists of brief descriptions of all of the algorithms and models used during experiments.

Chapter 5 describes the HS classification methodology and the course of action taken to address the research problems.

Chapter 6 focuses on the experimentation results, evaluation approach, models comparison and the analysis of the achieved results.

Chapter 7 summarizes the present work with concluding remarks, limitations, and includes an overview of the research work that can be expanded based on the current course of research.

Chapter 2

Background Definitions and Literature Review

To discuss the problem further we need to be familiarized with a few key data definitions relevant to the problem.

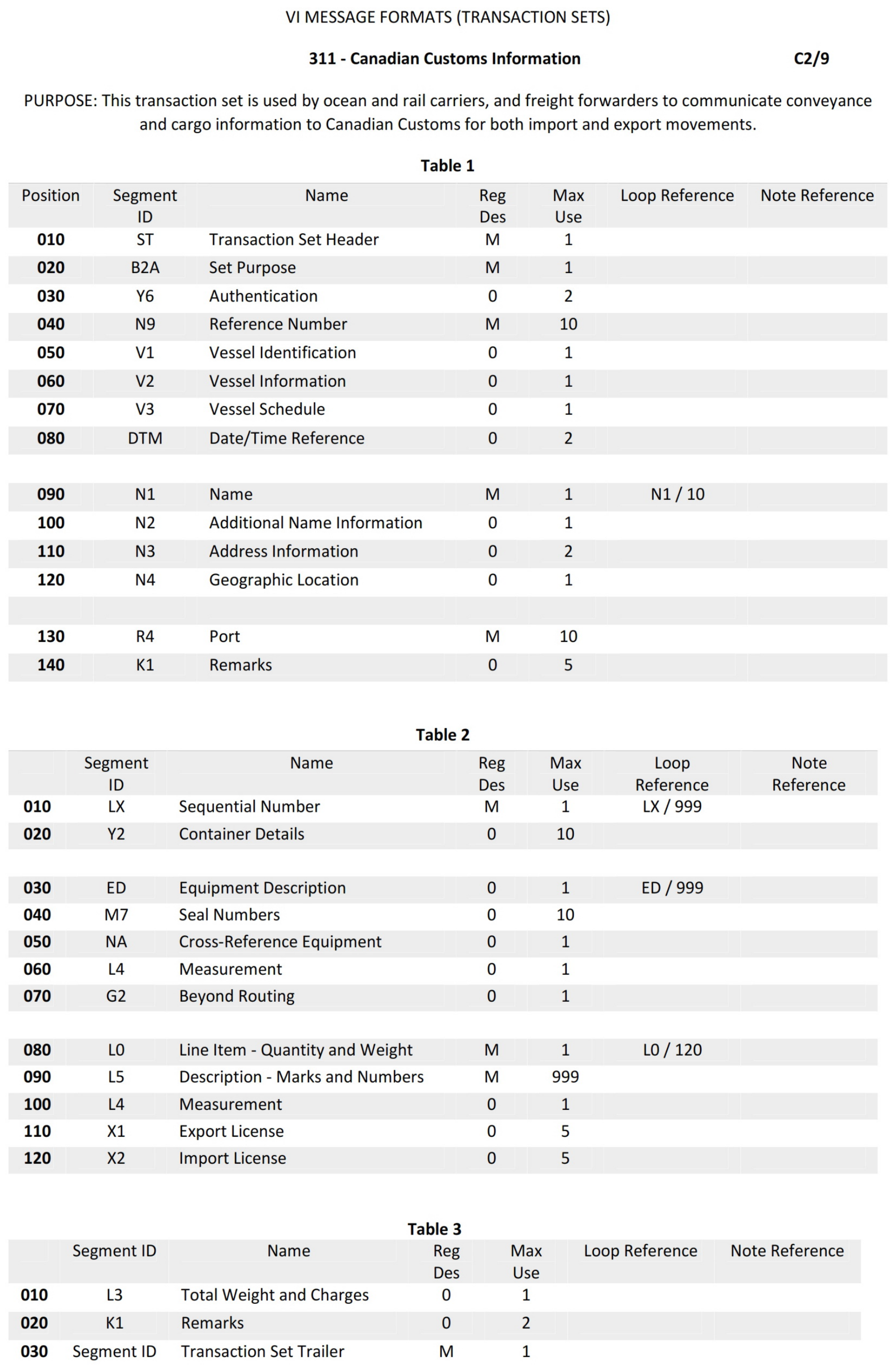
2.1 Definitions

We deal with 3 different types of entities when dealing with trade transactions: (1) the trade document sent by the shipper; (2) harmonized commodity code which is the primary identification factor for billing and checking of contents inside a container; and (3) United Nations defined commodity reference trade sheet, which contains harmonized commodity codes for each and every type of possible traded goods and is also the standard to validate when enclosing commodity codes in trade documents.

2.1.1 Trade Document

Trade documents are trade data documents or manifests usually written in EDI file format that designates different information related to a shipment such as date and time of shipment, trade and transaction numbers, vessel identification, shipper details and address, short commodity description, harmonized commodity code, etc. There are numerous EDI standards from EDI 200 to EDI 499 dealing with international trade, all using similar data definitions, with slight differences due to use or application. For instance, EDI 432 deals with the cost of space on rail cars in international shipping. An example diagram is shown in Figure 2.1 that describes the different types of possible tags in a 311 EDI standard. The figure also depicts how the Canadian Customs interpret the associated tags based on their respective data descriptions.

For our research we extensively use the commodity description field **L5**.



VI MESSAGE FORMATS (TRANSACTION SETS)

311 - Canadian Customs Information **C2/9**

PURPOSE: This transaction set is used by ocean and rail carriers, and freight forwarders to communicate conveyance and cargo information to Canadian Customs for both import and export movements.

Table 1

Position	Segment ID	Name	Reg Des	Max Use	Loop Reference	Note Reference
010	ST	Transaction Set Header	M	1		
020	B2A	Set Purpose	M	1		
030	Y6	Authentication	0	2		
040	N9	Reference Number	M	10		
050	V1	Vessel Identification	0	1		
060	V2	Vessel Information	0	1		
070	V3	Vessel Schedule	0	1		
080	DTM	Date/Time Reference	0	2		
090	N1	Name	M	1	N1 / 10	
100	N2	Additional Name Information	0	1		
110	N3	Address Information	0	2		
120	N4	Geographic Location	0	1		
130	R4	Port	M	10		
140	K1	Remarks	0	5		

Table 2

	Segment ID	Name	Reg Des	Max Use	Loop Reference	Note Reference
010	LX	Sequential Number	M	1	LX / 999	
020	Y2	Container Details	0	10		
030	ED	Equipment Description	0	1	ED / 999	
040	M7	Seal Numbers	0	10		
050	NA	Cross-Reference Equipment	0	1		
060	L4	Measurement	0	1		
070	G2	Beyond Routing	0	1		
080	L0	Line Item - Quantity and Weight	M	1	L0 / 120	
090	L5	Description - Marks and Numbers	M	999		
100	L4	Measurement	0	1		
110	X1	Export License	0	5		
120	X2	Import License	0	5		

Table 3

	Segment ID	Name	Reg Des	Max Use	Loop Reference	Note Reference
010	L3	Total Weight and Charges	0	1		
020	K1	Remarks	0	2		
030	Segment ID	Transaction Set Trailer	M	1		

Figure 2.1: This is the UN approved 311 EDI Standard

2.1.2 Harmonized Commodity Code

The Harmonized Item Description and Coding System (HS) is an international standard introduced in January 1988 and since maintained by the World Customs Organization (WCO) [22]. According to Rodolfo Vazquez, IMMEX Customs Manager and Foreign Trade Consultant, the importance of HS codes can be understood as, "Used on import (and export) declarations, HS codes identify the duty rates applicable to the specific goods, related to statistics, give regulators an opportunity to link Anti-Dumping Duties (ADD) and Countervailing Duties (CVD) to products, dictate how to qualify for preferential treatment, and can govern document and license requirements. Quite a laundry list—and that makes the correct HS code classification an important piece of information, especially when using an incorrect classification can lead to penalties and delays upon import [23]."

The HS codes system is currently signed by all members of the WCO (over 180 countries) and in use by nearly 200 countries. According to WCO, over 98% of goods in international trade are classified using HS codes [24]. As previously mentioned by the Customs Manager Rodolfo Vazquez, the codes make a number of things much easier to do. In simple terms they help customs authorities and cargo terminals apply import duties, collect trade statistics, work with trade agreements and control regulated goods. In shipping manifests Harmonized commodity code has a variety of ways of being written such as H.S.Code, HS Number, Commodity code, Commodity classification number, Tariff No. etc.

HS codes defined by the UN consist of 6 digits. However, we sometimes see HS Codes with more than 6 digits and that is because the universally agreed bit is only the first six digits. Countries are free to attach further digits and go up to even eight or ten digits HS codes with their own clarifications.

2.1.3 UN Comtrade Sheet

UN Comtrade sheet can be defined as United Nations approved commodity trade sheet for accepted goods. This is usually a tabular document most of the time represented in a Microsoft Excel worksheet which contains Harmonized commodity codes of every possible traded good between countries. Table 2.1 and 2.2 are examples of this document.

Classification	Code	Description	Code Parent	Level	isLeaf
H0	406	Cheese and curd	4	4	0
H0	40610	Dairy produce; fresh cheese (including whey cheese), not fermented, and curd	406	6	1
H0	40620	Dairy produce; cheese of all kinds, grated or powdered	406	6	1
H0	40630	Dairy produce; cheese, processed (not grated or powdered)	406	6	1
H0	40640	Dairy produce; cheese, blue-veined (not grated, powdered or processed)	406	6	1
H0	40640	Dairy produce; cheese (not grated, powdered or processed), n.e.s. in heading no. 0406	406	6	1
H0	407	Birds' eggs, in shell; fresh, preserved or cooked	4	4	0
H0	40700	Eggs; birds' eggs, in the shell, fresh, preserved or cooked	407	6	1
H0	408	Birds' eggs, not in shell; egg yolks, fresh, dried, cooked by steaming or boiling in water, moulded, frozen or otherwise preserved, whether or not containing added sugar or other sweetening matter	4	4	0
H0	40811	Eggs; birds' eggs, yolks, dried, whether or not containing added sugar or other sweetening matter	408	6	1
H0	40819	Eggs; birds' eggs, yolks, fresh, cooked by steaming or by boiling in water, moulded, frozen or otherwise preserved, whether or not containing added sugar or other sweetening matter	408	6	1

Table 2.1: A section of UN approved commodities trade sheet — 1

Classification	**Code**	**Description**	**Code Parent**	**Level**	**isLeaf**
H0	40899	Eggs; birds' eggs (not in shell, excluding yolks only), fresh, cooked by steaming or boiling in water, moulded, frozen, otherwise preserved, whether or not containing added sugar or other sweetening matter	408	6	1
H0	40899	Eggs; birds' eggs (not in shell, excluding yolks only), fresh, cooked by steaming or boiling in water, moulded, frozen, otherwise preserved, whether or not containing added sugar or other sweetening matter	408	6	1
H0	409	Honey; natural	4	4	0
H0	40900	Honey; natural	409	6	1
H0	410	Edible products of animal origin; not elsewhere specified or included	4	4	0
H0	41000	Animal products; edible, n.e.s. in this or other chapters	410	6	1
H0	5	Animal originated products; not elsewhere specified or included	TOTAL	2	0
H0	501	Human hair; unworked, whether or not washed or scoured; waste of human hair	5	4	0
H0	50100	Animal products; hair, human, unworked, whether or not washed or scoured, and waste of human hair	501	6	1
H0	502	Pigs', hogs' or boars' bristles and hair; and waste thereof	5	4	0
H0	50210	Animal products; hair and bristles, of pigs, hogs or boars, and waste thereof	502	6	1
H0	50290	Animal products; badger hair and other brush making hair and waste of such bristles or hair, n.e.s. in heading no. 0502 (excluding horsehair)	502	6	1
H0	503	Horsehair and horsehair waste; whether or not put up as a layer with or without supporting material	5	4	0
H0	50300	Animal products; horsehair and horsehair waste, whether or not put up as a layer with or without supporting material	503	6	1

Table 2.2: A section of UN approved commodities trade sheet — 2

2.2 Literature Review

Majority of the previous work is to help shippers correctly identify commodity codes for their goods to properly enclose them in a shipping manifest but our problem is to deal with situations where the shipping manifests are not following the proper standard defined by the manifest schema (such as Figure 2.1). Our objective is to handle scenarios when there are no commodity codes present in the trade manifest that can help the appropriate authorities in the billing of the goods in time. During the time of the research, we have not found any prior work done in this area that tries to help the cargo terminal's (ports, airports, railway and truck depot) side of things to process their transactions. However, after project conclusion there is one very recent and relevant work that was published in May, 2021 [25]. We will discuss about that particular research in detail and how our approach compares in a future Section. The overall literature review can be categorized into five categories based on their applicability as follows; 1) literature on cargo containers 2) literature on HS code recommendations for shippers 3) related literature in semantic analysis area 4) literature based on text classification approaches 5) most recent literature on HS code classification.

2.2.1 Literature on Cargo Containers

There has been a range of research regarding the contents of shipping containers, however much of the research today has been about optical imagery of the containers themselves [26], and RFID tags associated with the containers [27]. Data mining approaches have been used to study container contents requiring extra security [28]. Machine learning approaches have been used for anomaly detection [29]. Spatio-temporal data is also an important field of study [30], however, it is often studied using tweets or breaking news. Pattern matching of historical data has also been studied [31]. There has been prior work on using NLP to find geospatial information in text [32] and matching database schema [33]. What makes our problem more difficult is the text we are trying to match to shipping codes will be minimal and extremely tangled.

2.2.2 Literature On HS Code Recommendations For Shippers

From the shipper's side of things, we have found a few relevant works that are close to ours. Ontology-based system for the recommendation of HS Codes [34]. In order to perform the recommendation, the authors constructed ontologies and use semantic knowledge, matchmaking and reasoning mechanism. The background net approach for auto-categorization of HS Codes [15]. A background net, as suggested by the name, captures useful semantics of background information through incremental learning of co-occurrence of words in the text to achieve robust classification in specific application domain with open and evolving vocabulary. In 2019, Guo and Na Li proposed a text-image adaptive convolutional neural network to effectively utilize website information and facilitate the customs classification process [35]. Their proposed model includes two independent sub-models: one for text and the other for image.

2.2.3 Related Literature in Semantic Analysis Area

In terms of applicability, the following studies are close to ours but in different fields. The concept of semantic technique and studies regarding semantic analysis on semantic knowledge. Researchers spent much time and efforts on semantic net [36], and word sense disambiguation [37]. Hossain *et al.* showed the importance of semantics in the field of schema matching [38]. Semantic schema is an abstract structure that can be used to produce knowledge by proper interpretation. WordNet [39], Roget's Thesaurus [40], LDOCE [41] explored the concept of using existing lexical resources to automatically extract semantic knowledge. Nowadays techniques such as machine learning, statistical and algebraic approach proved achievements on approaches based on semantic knowledge.

2.2.4 Literature Based on Text Classification Approaches

HS code classification is essentially a text classification task. So, we can refer to a range of text classification techniques that we have performed in our methodology. Named Entity Recognition (NER) based approaches and the comparison of modern industry based NER softwares were studied [42]. Nath *et al.* performed a detailed survey on elastic search similarity algorithm [43]. The most commonly used machine learning (ML) based models for text classification include Naïve Bayes [44, 45], Semantic Random Forest [46, 47], Support Vector Machine (SVM) [48, 49, 50]. Information retrieval techniques based on word and sentence embedding have also been explored [51, 52]. Liga *et al.* achieved encouraging scores in argumentative evidence classification and showed the superiority of transfer learning (TL) with sentence embeddings for text classification [53]. Researchers at Google found that transfer learning using sentence embeddings tends to outperform word level transfer [54].

2.2.5 Most Recent Literature on HS Code Classification

Thus far the most relevant and very recent (May, 2021) work in this research area was published from researchers of Beijing Jiaotong University, Beijing Key Laboratory of Traffic Data Analysis & Mining and CAAC Key Laboratory of Intelligent Passenger Service of Civil Aviation, Beijing, China [25]. The research proposed an HS code classification neural network named 'HScodeNet' by incorporating the hierarchical sequential and global spatial information of texts, in which a hierarchical sequence learning module is designed to capture the sequential information and a text graph learning module is designed to capture the spatial information of commodity description texts [25]. HScodeNet is hosted publicly and works in Chinese language. Thus, a comparison with our approach was difficult due to the language barrier. However, even after the conclusion of our project we tried to compare our approach with HScodeNet in limited capacity. We found that ours is a safer and better approach for the given trade data, which is mostly in English with some other languages, and it is both unstructured and very noisy without any sort of hierarchical structure elements present. We show the detailed comparison in Section 6.6.

2.3 Knowledge Gap

In this section, we will refer with a general term 'regulatory entities' to organizations such as customs offices, inspections, and other government or international organizations overseeing and regulating international transportation of goods. A majority of the previous work in this area is on shipping containers themselves rather than their contents or to help shippers correctly identify commodity codes before the trade. However, our task at hand is to deal with situations where the trade documents are not following the proper standard. Cases where there are no commodity codes present in the trade document that can help regulatory entities in the billing of the goods in time. Our problem's difficulty is in the text describing the product being shipped within the trade document descriptions which is often handwritten then typed in an EDI file. This short text to describe a commodity is often noisy, extremely tangled and with multiple variations. It is very difficult to automatically verify product names without HS-code for the current Information system of inspection and custom authorities. Several regulatory entities do not offer a descriptive standard for traders to properly enclose HS-code while they have reported that in many cases, it is a very tiresome and difficult job to identify relevant code from the product description of an EDI trade manifest. This lack of standardization creates complications, inconsistency and inefficiency in the process of inspection. During the time of this research, we did not find any prior work done in this area that tries to address the regulatory entities's side of things. So, the knowledge gap in this area was significant to address.

Chapter 3

Research Problem and Datasets

3.1 Research Problem

After conducting the literature survey, it was certain that the techniques available out there are mostly to help shippers identifying HS codes for different products to prepare the shipping manifests. There wasn't anything that's particularly helpful for the entities that need to read and process the shipping manifests to classify or correct commodity codes for a manifest. Thus resulting in various risks discussed in Section 1.1.

The diversity in the initial reason for HS code classification comes with different parameters for success. For example, an auto-classification tool can solve many challenges for an e-commerce retailer such as immediate and responsive returns, high quantity of items are auto classified instantly. However, accuracy can be a challenge since importers can not afford a lack of accuracy. For example, the classifier determines if the imported product is subject to ADD (Anti-Dumping Duties), is heavily restricted from a license perspective, or is subject to quotas. One single missclassification during auto-classification can cost someone millions of dollars.

There are three key components to successful auto-classification other than a decent amount of human expert's classification expertise. The three key components are: commodity description, the classification logic, and the classification reference database.

First, commodity description. Commodity description is the primary identification factor for the traded commodity. Poor descriptions, lack of detail, or even incorrect specifications will likely lead to an incorrect HS code with all related consequences. These descriptions are usually short, noisy and extremely tangled, so sufficient attention is required for correct parsing of the parts identifying the commodity being traded.

Second, the classification logic. Whether the classification is assigned by a person or a classifying tool, the classification logic can not lack logic. This can mean a number of things such as: rule-based classification that guides the correct classification in a decision matrix; the ability to ignore irrelevant information in commodity description (e.g., color, weight); the ability to observe characteristics that may be needed in one case but not for another (e.g., fresh fruits vs dried fruits), including item compositions that are usually very important. The logic must also ensure a way to 'smart search', or search across the entire commodity reference database to generate results from. It is necessary to explore transfer learning (TL) using advanced sentence embedding techniques to develop solutions and HS-classification system to recognize HS-codes intelligently, based on the simplified commodity name.

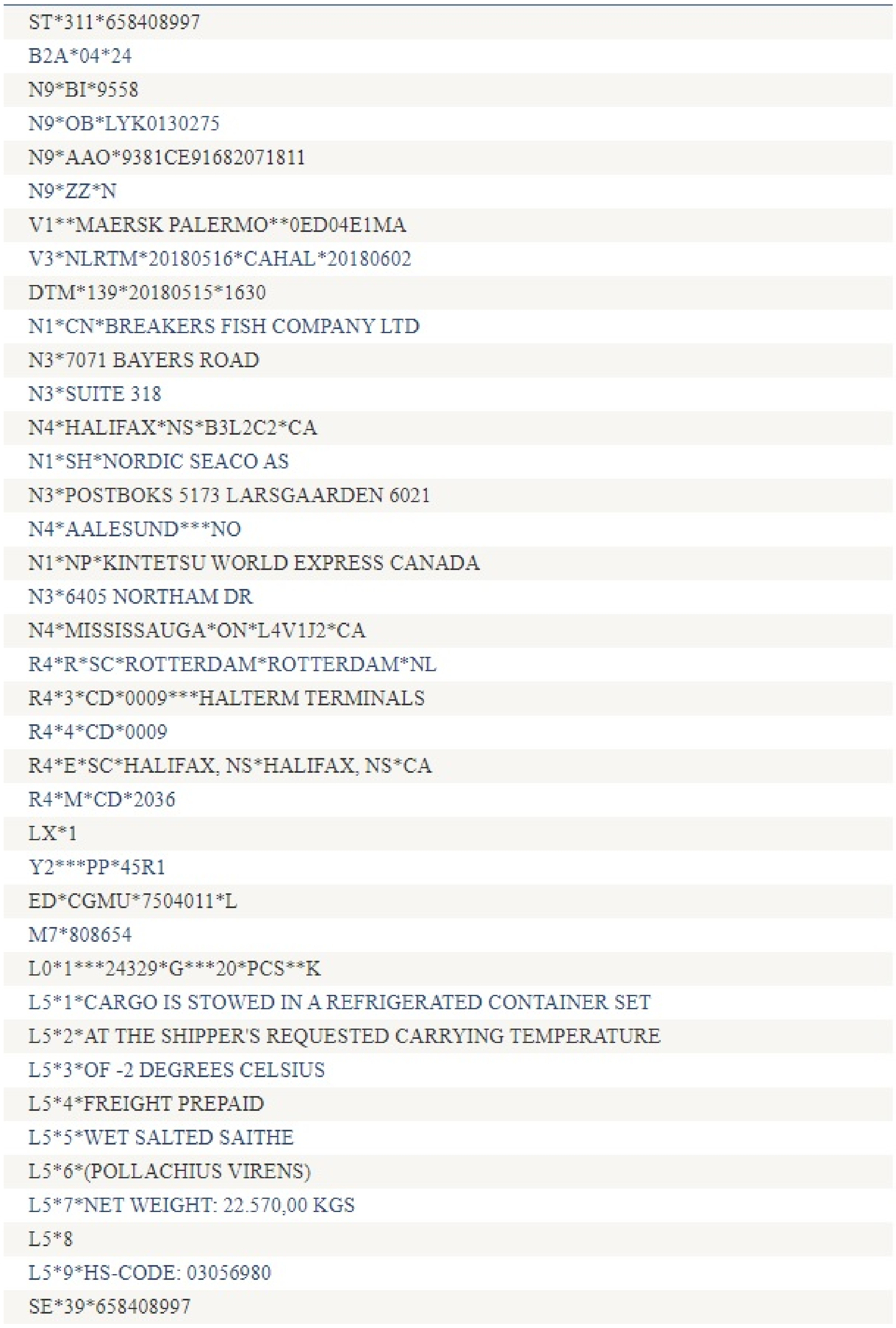
Third, the classification reference database. The classification logic must match a description with an HS code by matching it with a trade document's commodity description with the same semantic meaning and for better context a natural language reference. This may include a trade document reference or information gained from past classification repositories of similar products. Regardless, all references need to be reviewed by a human expert before the final classification is determined. The logic is only as firm as the foundation on which it is established. Fortunately, an excellent, up-to-date reference file is available on the UN website.

3.2 Data Sources and Datasets

The details about the data used in the research, the EDI standards or the provider of the data can not be disclosed due to IP and legal reasons. However, the definitions of each are already discussed in section 2.2.

An example of an open source EDI trade document is shown in Figure 3.1. The documents used for research were much more complex and larger in size.

Our data varied for each major shipping and transport corporation. For example, the sample EDI is for a container aboard the ship MAERSK PALERMO (V1 tag identifies vessel information) which is part of the MAERSK LINE company [55, 56]. Another example of a different vessel would be the recently stuck container ship in the Suez canal from EVERGREEN [57]. Although MAERSK and EVERGREEN



```
ST*311*658408997
B2A*04*24
N9*BI*9558
N9*OB*LYK0130275
N9*AAO*9381CE91682071811
N9*ZZ*N
V1**MAERSK PALERMO**0ED04E1MA
V3*NLRTM*20180516*CAHAL*20180602
DTM*139*20180515*1630
N1*CN*BREAKERS FISH COMPANY LTD
N3*7071 BAYERS ROAD
N3*SUITE 318
N4*HALIFAX*NS*B3L2C2*CA
N1*SH*NORDIC SEACO AS
N3*POSTBOKS 5173 LARSGAARDEN 6021
N4*AALESUND***NO
N1*NP*KINTETSU WORLD EXPRESS CANADA
N3*6405 NORTHAM DR
N4*MISSISSAUGA*ON*L4V1J2*CA
R4*R*SC*ROTTERDAM*ROTTERDAM*NL
R4*3*CD*0009***HALTERM TERMINALS
R4*4*CD*0009
R4*E*SC*HALIFAX, NS*HALIFAX, NS*CA
R4*M*CD*2036
LX*1
Y2***PP*45R1
ED*CGMU*7504011*L
M7*808654
L0*1***24329*G***20*PCS**K
L5*1*CARGO IS STOWED IN A REFRIGERATED CONTAINER SET
L5*2*AT THE SHIPPER'S REQUESTED CARRYING TEMPERATURE
L5*3*OF -2 DEGREES CELSIUS
L5*4*FREIGHT PREPAID
L5*5*WET SALTED SAITHE
L5*6*(POLLACHIUS VIRENS)
L5*7*NET WEIGHT: 22.570,00 KGS
L5*8
L5*9*HS-CODE: 03056980
SE*39*658408997
```

Figure 3.1: Example of an EDI trade document [2]

deal with the same EDI standard, their way of writing commodity descriptions in the trade document is different. So, I had to come up with individual data cleaning approaches specific to these different shipping companies. Although the given data was from seven different shipping companies, the data for research was only sufficient from four of them. Due to reasons of confidentiality, we will later identify these shipping companies as Shipper 1 – 4. At the initial stage of the HS code classification methodology, different approach explorations were conducted using only Shipper 1's data. The reason being, Shipper 1 had the high quality data, which means most of the EDI trade documents from this shipping company are with properly formatted tags.

Chapter 4

Underlying Algorithms and Models

In this chapter, I will provide a review of the algorithms and models that have been explored during different approaches. We can categorize these models based on their working principles.

4.1 Named Entity Recognition

Named Entity Recognition (NER) is a subtask of information retrieval that aims to recognize mentions of rigid designators from text belonging to predefined semantic types such as person names, organizations, locations, medical codes, products, quantities, monetary values, percentages, etc. [58]. NER is used in many fields in Natural Language Processing (NLP), and it can help to answer real-world questions, such as: "How many people are involved in this news article?", "What product was mentioned in the tweet?", "Does the tweet contain the product's location?", and similar. To evaluate the quality of a NER system, there are a few different measures that have been defined. The most common measures are called Precision, Recall, and F1 score.

Notable NER platforms include: OpenNLP [59], StanfordNER [60], and SpaCy [61]. We used SpaCy to implement Named Entity Recognition.

4.1.1 SpaCy

SpaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. I chose SpaCy as my NER model because it is designed specifically for production use. SpaCy features fast statistical NER as well as an open-source named-entity visualizer. A typical workflow of a NER model is as follows:

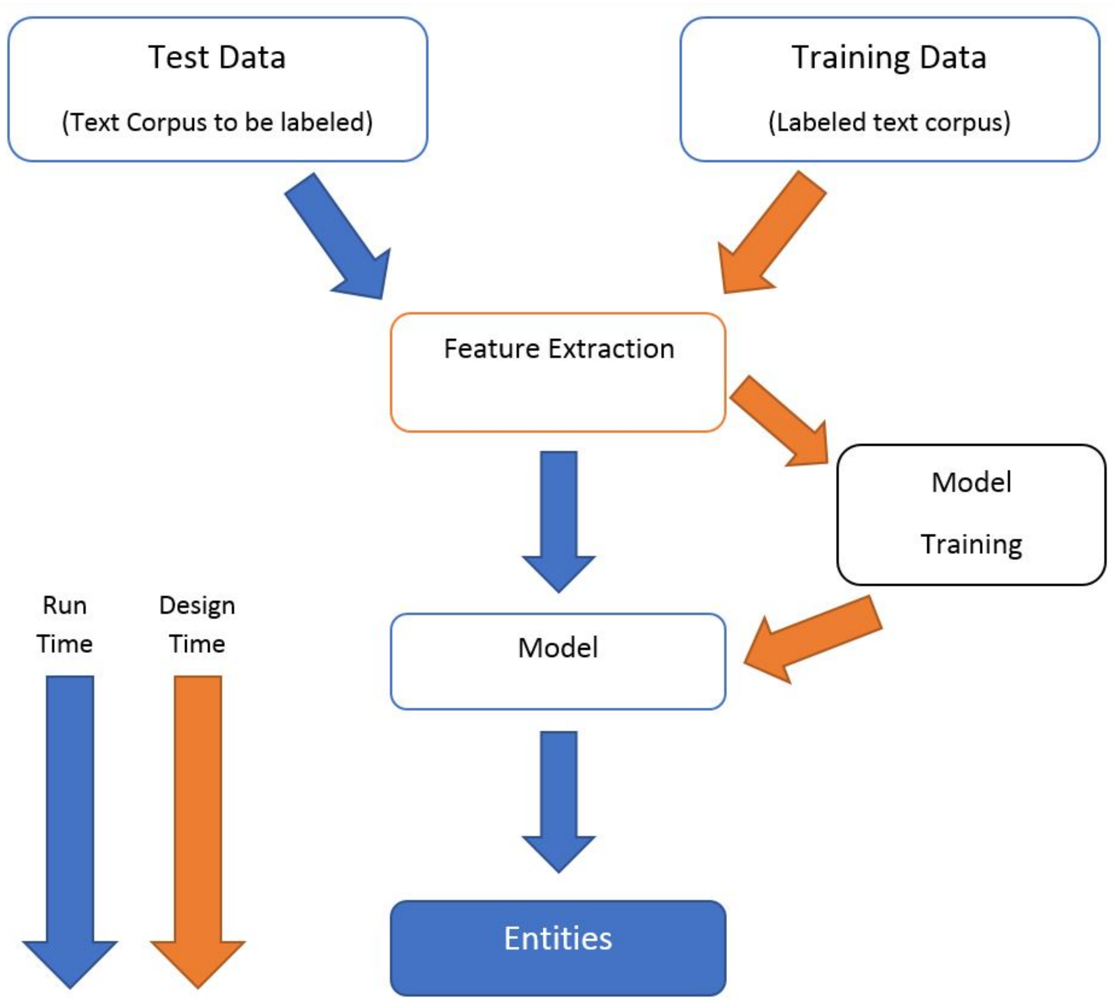


Figure 4.1: Workflows for Named Entity Recognition [3]

4.2 Elastic Search

Elasticsearch is a distributed, open-source search and analytics engine built on Apache Lucene and developed in Java [43]. Elasticsearch allows us to store, search, and analyze huge volumes of data quickly and in near real-time and give back answers in milliseconds. It is able to achieve fast search responses because instead of searching the text directly, it searches an index. Elasticsearch uses a structure based on the documents instead of tables and schemas. It comes with extensive REST APIs for storing and searching the data. It can be used to search all kinds of data.

4.3 Supervised Machine Learning Algorithms

Supervised learning can best be described as the concept of function approximation. We train an algorithm and we choose the function that best describes the input data, the one that for a given input X makes the best estimation of output y.

$$X \rightarrow y \tag{4.1}$$

Supervised learning algorithms try to model relationships and dependencies between the input features and the target prediction output.

4.3.1 Multinomial Naïve Bayes

Naïve Bayes is computationally very efficient and easy to implement, it is one of the most popular supervised learning classifications that is used for the analysis of the categorical text data [44]. Often, it is used as a baseline in text classification. Naïve Bayes is based on Bayes' theorem, where the adjective Naïve states that features in the data set are mutually independent. The occurrence of one feature does not affect the probability of occurrence of the other. In experiments with small sample sizes, Naïve Bayes can outperform the most powerful alternatives. Two event models are commonly used:

- Multi-variate Bernoulli model
- Multinomial Naïve Bayes

Empirical comparisons provide evidence that the multinomial model tends to outperform the multi-variate Bernoulli model if the vocabulary size is relatively large [62]. Due to the robustness, being easy to implement, fast, and accurate, it is used in many different fields. For example, spam filtering in email, diseases diagnosis, decisions about treatment, RNA sequence classification in taxonomic studies, etc.

4.3.2 Decision Tree (DT)

A decision tree is used for classification and regression problems [63]. The decision tree is a structure consisting of nodes that form a directed tree with a node called "root" that has no incoming edges. Every other node has exactly one incoming edge.

The node with outgoing edge is called a test node. Every other node is called a leaf node. Leaf nodes represent a final classification or decision. The decision tree-based algorithms build the trees based on the principle of information entropy. When the tree is under construction, the normalized information gain is calculated for individual nodes. The feature with the highest value of information gain is chosen for making decisions. The tree works recursively in the same manner till the end of classification.

A major advantage of decision tree is that it forces the consideration of all probable outcomes of a decision and traces each path to it's conclusion. It analyses the consequences along each branch and identifies decision nodes that need further analysis. For a simple decision tree the model is easy to interpret. However, when the tree is designed to perfectly fit all samples in the training data, it ends up with branches with strict rules of sparse data which effects the accuracy when predicting samples that are not part of the training set and causing over-fitting. Pruning the branches of the tree can be performed to address over-fitting.

4.3.3 Random Forest

A common disadvantage of decision trees is they are prone to overfitting. A Random Forest is an ensemble of decision trees [46]. Each tree in the random forest makes a class prediction and the class with the most votes becomes the model's final prediction. The process of forest generation is based on the principle of collecting the trees with controlled variance.

There are a few advantages of Random Forest. A low number of model parameters makes it resistant to overfitting. While the number of trees increases in a random forest, the variance of the model decreases without affecting the bias. The random forest has a few disadvantages as well. For example, the forest is difficult to interpret and its dependent on a random generator.

4.3.4 Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a popular supervised learning algorithm used for pattern recognition and regression analysis, which later found its application in the field of text classification [64, 65]. SVM classifier is based on finding an optimal hyperplane which distinguishes the entities belonging to two classes as positive or negative.

The SVM algorithm uses mathematical functions defined as Kernels. For some classification problems, it is not always possible for SVM to find a hyperplane or a linear decision boundary but if the data is projected into a higher dimension from the original space, it may be possible to get a hyperplane in the projected dimension that helps to classify the data. The advantages of SVM include effectiveness in high dimensional spaces, memory efficiency, it is also effective where the number of dimensions is greater than the number of samples.

SVM is majorly used for performing binary classification, but it can also be used for multi-class classification by finding the optimal hyperplane between each pair of classes. In the case of commodity code classification, I used multicode classification since similar descriptions can lead to almost similar codes.

4.4 Semantic Textual Similarity (STS)

Semantic Textual Similarity is an algorithm that enables AI systems to be able to understand semantically similar queries. These algorithms aim to not only improve the quality of responses of services such as Google Assistant but also make these interactions feel more natural as well. Semantic textual similarity deals with determining how similar two pieces of texts are. For example, if the user asks “How old are you?” or “What is your age?”, they will expect the same response.

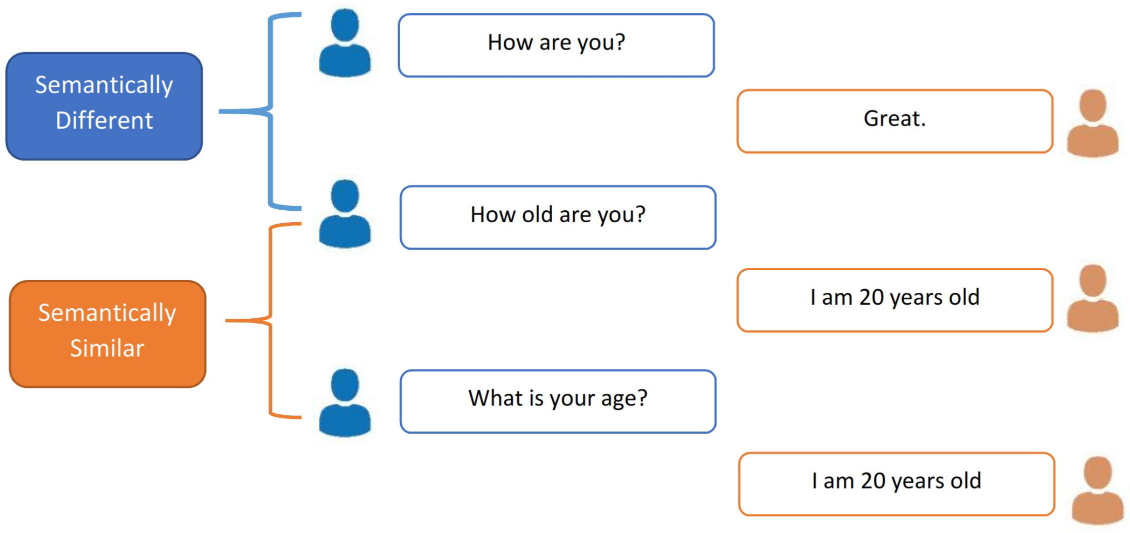


Figure 4.2: Example of Semantic Textual Similarity (STS) [4]

In order to understand semantic textual similarity (STS) matching, it is ideal to be familiar with these four technical concepts in this domain, specific to the research problem.

4.4.1 Embeddings

The perceptive human brain is capable of comprehending humor, sarcasm, sentiment, and far more, very easily for a given sentence. For a machine to comprehend the meaning of a text, it is important that we represent this text in a language that the machine can understand. Word embedding is a learned representation for text where words that have an equivalent meaning have a similar representation [66]. Generating word embeddings with a really deep architecture is just too computationally expensive for an outsized vocabulary. This is often the primary reason why it took until the year 2013 for word embeddings to explode onto the NLP stage [67].

4.4.2 Word vs. Sentence Embeddings

Word embeddings can represent the meaning of the words in a text. Sometimes we need to go beyond just the meaning of words in a text and encode the meaning of the whole sentence to be able to understand the context in which the words are said. In our problem with very short and tangled texts, sentence embeddings capture much more context about the commodity description than word embeddings. Suppose, we come across a sentence like ‘pet food products’, and a few sentences later, we read ‘cat and dog food’. How can we make the machine draw the inference between ‘pet’ and ‘cat and dog’?

Clearly, word embedding would fall short here, and thus, we use Sentence Embedding [54]. Sentence embedding techniques represent entire sentences and their semantic information as vectors. This helps the machine in understanding the context, intention, and other nuances in the entire text.

4.4.3 Transfer Learning

Transfer Learning (TL) is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [53]. For example, knowledge gained while learning to recognize bicycles could be applied when trying to recognize motorbikes. The principle idea behind transfer learning can be shown in Figure 4.3. Transfer learning using sentence embeddings tends to outperform word-level transfer [54]. We use deep transfer learning for natural language processing using sentence transformers to produce sentence-level embeddings.

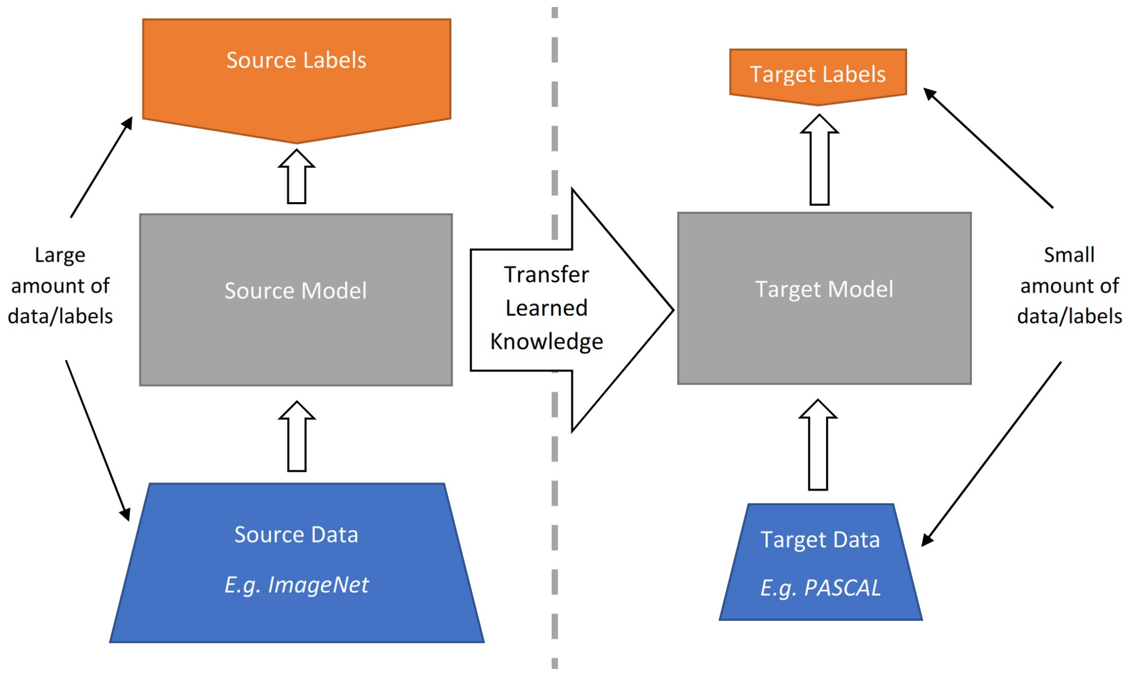


Figure 4.3: The idea behind Transfer Learning (TL)

4.4.4 Cosine Similarity

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size [68]. Mathematically, it measures the cosine of the angle between two vectors in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, the higher the cosine similarity.

$$similarity(A, B) = \cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.2}$$

4.5 Pre-Trained Semantic Textual Similarity (STS) Models

A pre-trained model is a model created and trained by someone else to solve a similar problem. In practice, that "someone" is almost always a tech giant or a group of reputed researchers. Usually a very large dataset is chosen as their base datasets, such as ImageNet or the Wikipedia Corpus. Then, large neural network (e.g., VGG19 has 143,667,240 parameters) is created to solve a particular problem (e.g., image classification for VGG19). Of course, this pre-trained model must be public so that we can access and repurpose it.

The above is true for the following STS models invented by researchers from Google and Facebook. Some of the base versions of most common STS models are: Universal Sentence Encoder (USE), Sentence Bert (Sbert), Infersent. It is otherwise computationally and economically very expensive to develop a semantic textual similarity model from scratch, so we resort to using the pre-trained versions.

4.5.1 Universal Sentence Encoder

The Universal Sentence Encoder (USE) developed by Google Research encodes text into high dimensional vectors that can be used for text classification, semantic similarity, clustering and other natural language tasks [54].

The model is trained and optimized for greater-than-word length text, such as sentences, phrases or short paragraphs. As stated in the academic research document, the unsupervised training data for the sentence encoding models are drawn from a variety of web sources such as Wikipedia, web news, web question-answer pages and discussion forums (Figure 4.4). The input is variable-length English text and the output is a 512 dimensional vector. The primary goal behind the development of USE is to encode sentences into embedding vectors that specifically target transfer learning (TL) to various NLP tasks like sentiment analysis, text classification, sentence similarity, etc.

There are two models presented for producing sentence embeddings that demonstrate good transfer to a number of other of other NLP tasks and we can use either of the two: (a) Transformer, and (b) Deep Averaging Network (DAN). While the one with the Transformer architecture has higher accuracy, it is computationally more intensive. The one with DAN architecture is computationally less expensive and with little lower accuracy. Both of these sentence encoding models are implemented in TensorFlow and made publicly available on TensorFlow Hub.

These models take English strings as input and produce a fixed dimensional embedding representation of the strings as output. The following is the basic flow of USE:

1. Tokenize the sentences after converting them to lowercase.

2. Depending on the type of encoder, the sentence gets converted to a 512-dimensional vector

    - If we use the transformer, it is similar to the encoder module of the transformer architecture and uses the self-attention mechanism.
    - The DAN option computes the unigram and bigram embeddings first and then averages them to get a single embedding. This is then passed to a deep neural network to get a final sentence embedding of 512 dimensions.

3. These sentence embeddings are then used for various unsupervised and supervised tasks like Skipthought and Natural Language Inference (NLI) [69, 70]. The trained model is then again reused to generate a new 512 dimension sentence embedding.

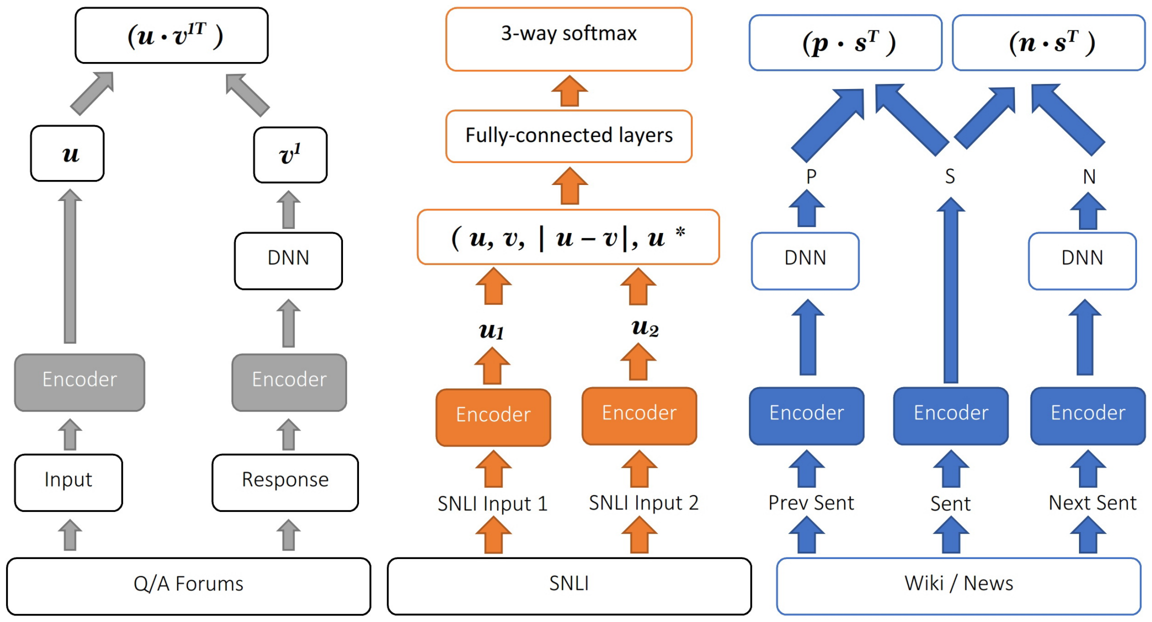


Figure 4.4: Multi-task training in USE. A variety of tasks and task structures are joined by shared encoder layers or parameters [4]

For our task we use the pre-trained DAN version of Sentence Encoder with a pre-processed trade document description as a query and all of the product descriptions from UN comtrade sheet as a corpus. We then use USE to find similar sentences surrounding the query based on cosine similarity scores.

4.5.2 Infersent

InferSent from Facebook is an interesting approach by the simplicity of its architecture. It is an NLP technique for universal sentence representation that uses supervised training to produce high transferable representations [5]. Infersent uses the Stanford Natural Language Inference (SNLI) Corpus (a set of 570k pairs of sentences labeled with 3 categories: neutral, contradiction, and entailment) to train a classifier on top of a sentence encoder.

A basic operational flow of Infersent can be seen from the following Figure 4.5:

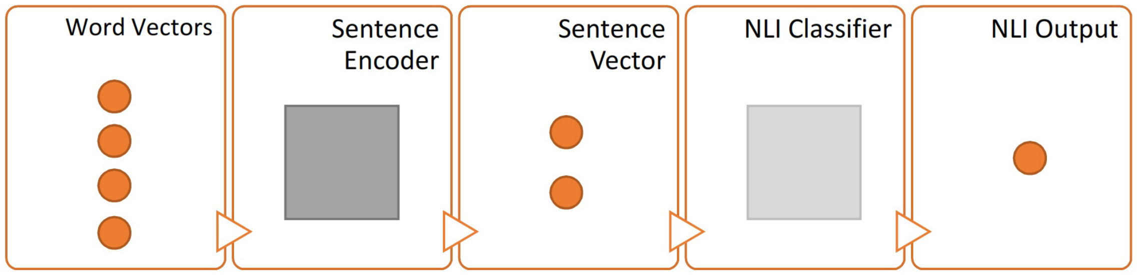


Figure 4.5: General flow of InferSent [5]

The architecture has two important parts:

1. first, is the sentence encoder that takes word vectors and encodes sentences into vectors.

2. second, the NLI classifier that takes the encoded vectors in and outputs a class among entailment, contradiction and neutral.

Sentence Encoder: The paper discusses multiple architectures for sentence encoding such as BiLSTM with max or mean pooling, LSTM and GRU, Self Attentive network and Hierarchical Convolutional Network [5]. They show that Bi-directional LSTM with max or mean pooling works best for sentence encoding.

NLI Classifier: After the sentence vectors are fed as input to this model, 3 matching methods are applied to extract relations between the premise text, u and hypothesis v as shown in Figure 4.6:

$$\textit{concatenation of the two representations: } (u, v) \tag{4.3}$$

$$\textit{element-wise product: } |u * v| \tag{4.4}$$

$$\textit{and, absolute element-wise difference: } |u - v| \tag{4.5}$$

The resulting vector captures information from both the text, u, and the hypothesis, v, and is fed into a 3-class classifier consisting of multiple fully connected layers followed by a softmax layer [5].

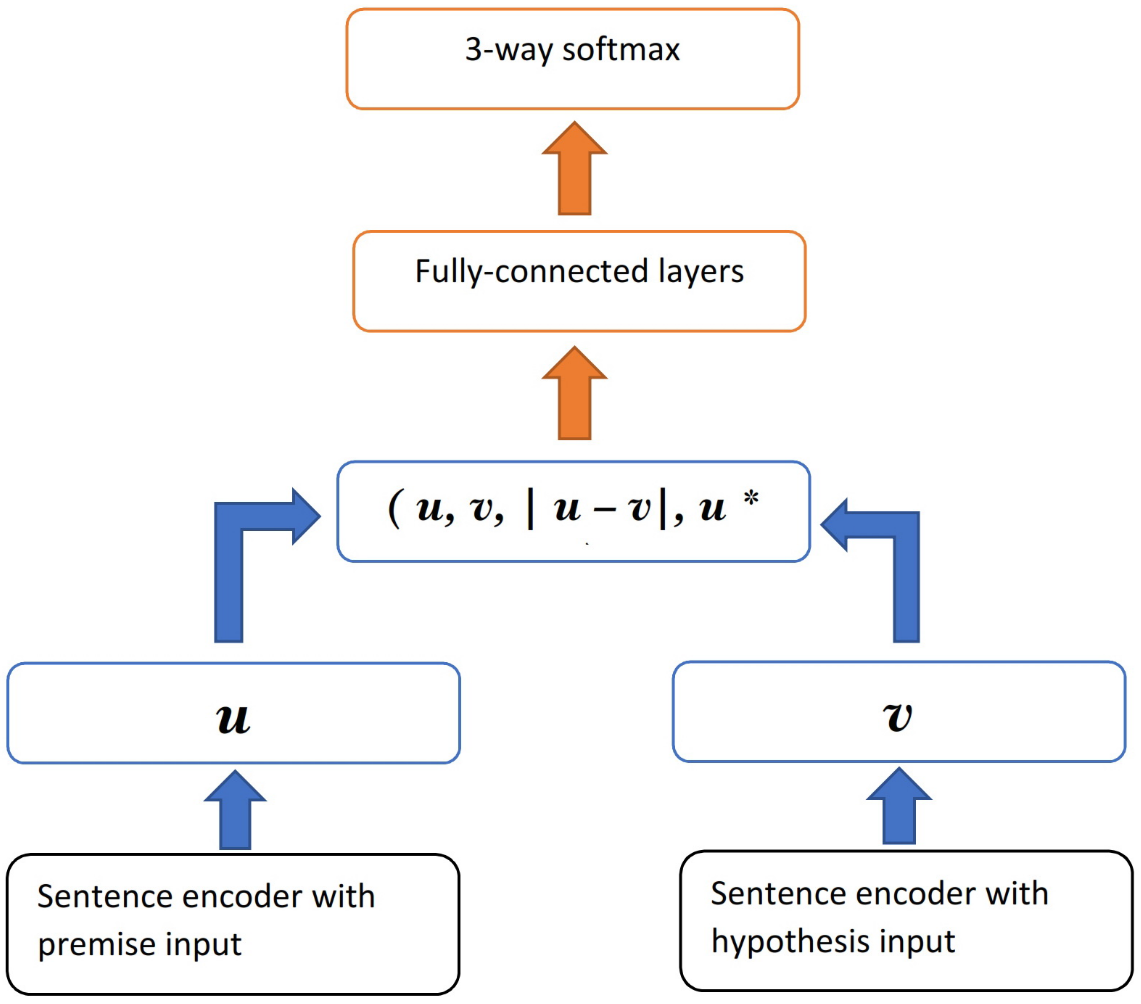

3-way softmax
Fully-connected layers
(u, v, | u – v|, u *
u
v
Sentence encoder with premise input
Sentence encoder with hypothesis input

Figure 4.6: NLI Training Scheme [5]

The paper concludes that an encoder based on a bi-directional LSTM architecture with max pooling, trained on the Stanford Natural Language Inference (SNLI) dataset, provides state-of-the-art sentence embeddings compared to all existing alternative unsupervised approaches like SkipThought or FastSent, while being much faster to train [5]. Another important feature is that the first version of InferSent uses GloVe vectors for pre-trained word embeddings [71]. A more recent version of InferSent, known as InferSent2 uses fastText [72].

4.5.3 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a large number of NLP tasks [6]. The model has been pre-trained on Wikipedia and BooksCorpus and requires task-specific fine-tuning [73]. Bert architecture is deeply bidirectional, meaning that BERT learns information from both the left and the right sides of a token's context during the training phase.

The bidirectionality of a model is very important for understanding the true meaning of a language. Let us illustrate the importance of bidirectionality using Figure 4.7. There are two sentences in this example and both of which involve the word "bank":

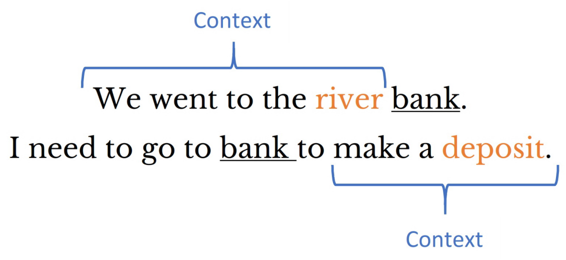

Context
We went to the river bank.
I need to go to bank to make a deposit.
Context

Figure 4.7: BERT captures both the left and right context [6]

If we try to interpret the predict of the word "bank" by only taking into consideration either the left or the right context, then we are at an error in at least one of the two given sentences. One way to deal with this is to consider both the left and the right context before making a prediction. That is exactly what BERT does [74].

The BERT architecture builds on top of Transformer. The paper introduces two variants, BERT Base with 12 layers (transformer blocks), 12 attention heads, and 110 million parameters. BERT Large with 24 layers (transformer blocks), 16 attention heads and, 340 million parameters. BERT is pre-trained on two NLP tasks, Masked Language Modeling and Next Sentence Prediction. After the initial release of BERT, several other versions were introduced each of which have their edge on different tasks. We use most of these Bert variations to find out which works best for our trade data.

4.5.4 DistilBERT

The dominance of Transfer Learning (TL) in Natural Language Processing (NLP) from large-scale pre-trained models also made it challenging to operate these large models efficiently due to constrained computational training or inference budgets. Addressing such concerns, DistilBERT is a pre-trained smaller general-purpose language representation model, which can also be fine-tuned with good performances on a wide range of tasks like its larger counterparts [75]. While most prior work investigated the utilization of distillation for building task-specific models, DistilBERT leverages knowledge distillation during the pre-training phase and shows that it is possible to to scale back the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. To leverage the inductive biases learned by larger models during pre-training, the authors of the paper introduce a triple loss combining language modeling, distillation and cosine-distance losses [75]. As a result, this smaller, faster and lighter model is much cheaper to pre-train.

4.5.5 RoBERTa

Robustly optimized BERT approach aka RoBERTa from Facebook AI is an optimisation of Google's popular BERT system for pre-training Natural Language Processing (NLP) systems. As stated by Facebook AI,"RoBERTa builds on BERT's language masking strategy, wherein the system learns to predict intentionally hidden sections of text within otherwise unannotated language examples" [76]. RoBERTa was implemented in PyTorch. Some of the key modifications from BERT includes:

1. Training the model longer, with bigger batches and over more data.
2. Training on longer sequences.
3. Removing BERT's next-sentence prediction (NSP) task.
4. Introducing dynamic masking so the masked token changes during the training epochs.

5. The addition of CommonCrawl News dataset [77].

This allows RoBERTa to outperform BERT on the masked language modeling objective and on GLUE benchmark results. The General Language Understanding Evaluation benchmark (GLUE) is a collection of datasets used for training, evaluating, and analyzing NLP models relative to one another [78].

4.5.6 XLM-RoBERTa-Multilingual

XLM-R or XLM-RoBERTa-Multilingual from FacebookAI shows that pretraining multilingual language models at scale lead to significant performance gains for a wide range of cross-lingual transfer tasks. The authors train a Transformer-based masked language model on one hundred languages, using more than two terabytes of filtered CommonCrawl data. “The model significantly outperforms multilingual BERT (mBERT) on a variety of cross-lingual benchmarks, including +13.8% average accuracy on XNLI, +12.3% average F1 score on MLQA, and +2.1% average F1 score on NER” [79]. The performance of XLM-R is particularly effective on low-resource languages, improving 11.8% in XNLI accuracy for Swahili and 9.2% for Urdu over the previous XLM model [80]. The authors also present a detailed empirical evaluation of the key factors that are required to achieve these gains, including the trade-offs between (1) positive transfer and capacity dilution and (2) the performance of high and low resource languages at scale. Finally, it’s shown that for the first time, it is possible for a multilingual model to not sacrifice per-language performance.

4.5.7 Sentence-BERT

The downfall of computing sentence similarity using BERT is that both sentences are required to be fed into the network causing a massive computational overhead. If we use BERT for a sentence similarity task, finding the most similar pair in a collection of 10,000 sentences requires about 50 million inference computations ($\approx$ 65 hours).

The architecture of BERT makes it unsuitable for semantic similarity search as well as for unsupervised tasks like clustering. As stated in the academic paper, “Sentence-BERT (SBERT) is a modification of the pre-trained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity” [7]. SBERT can reduce the

65 hours of BERT or RoBERTa as mentioned above to 5 seconds (calculating cosine similarity takes about 0.01 seconds) while maintaining the accuracy from BERT or RoBERTa. The model uses combined SNLI (Stanford Natural Language Inference) and Multi-Genre NLI data sets during training. SBERT was introduced in 2018 and immediately took the top spot for sentence embeddings. The architecture has 4 key concepts:

1) Attention
2) Transformers
3) BERT
4) Siamese Network

First, Siamese Network like architecture is used to provide 2 sentences as input [81]. Afterwards, these 2 sentences are passed to BERT models and a pooling layer to generate their sentence embeddings. Next, the embeddings for the pair of sentences work as inputs to calculate the cosine similarity. Similarly, Sentence-BERT also takes a sentence and corpus of sentences as input and then lists the top k similar sentences based on the order of cosine similarity. The architectures of SBERT for sentence similarity and classification tasks are displayed in Figure 4.8.

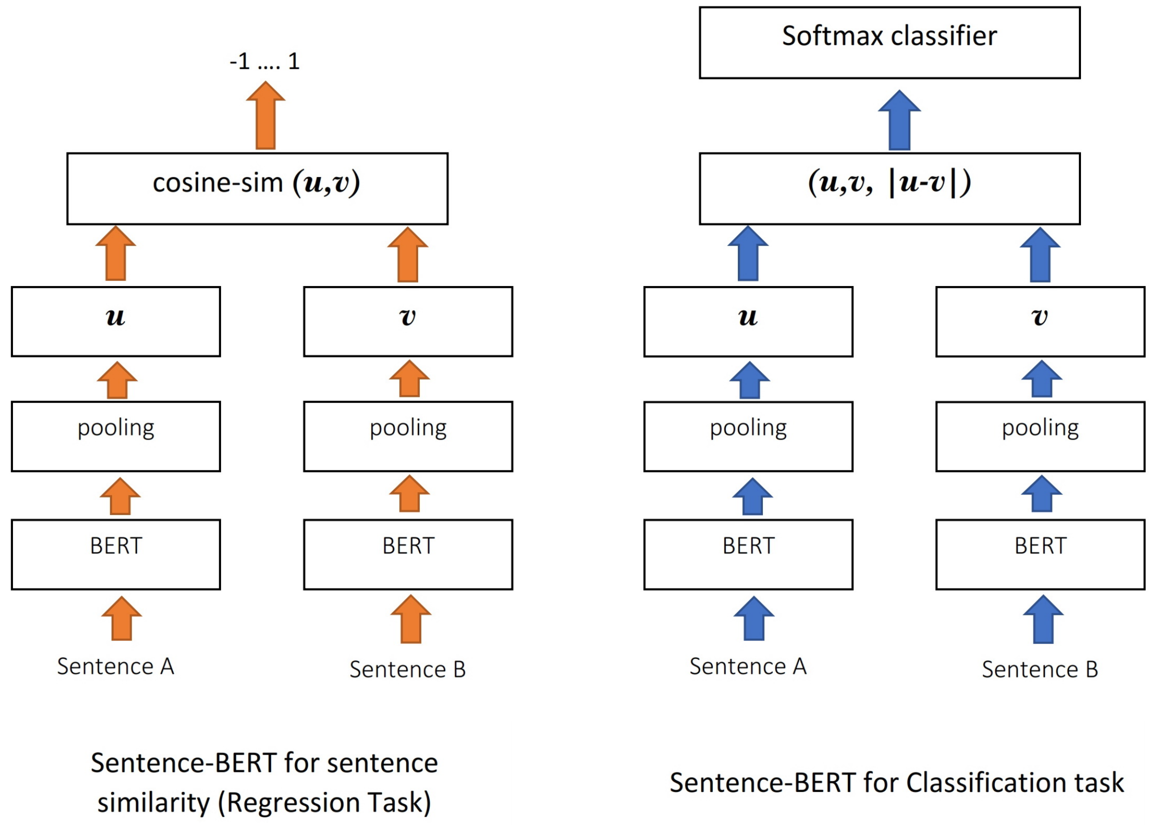


Figure 4.8: Sentence-BERT architectures [7]

To summarize, in this chapter we have discussed several algorithms and models with their working principles. Next chapter will use these concepts in our HS Code classification methodology.

Chapter 5

HS Code Classification Methodology

The proposed methodology consists of three different steps each leading to the next:

1. EDI Parsing

2. Commodity description pre-processing, and

3. HS code classification

Due to the nature of the knowledge gap discussed above, these steps are novel approaches built to clean trade documents. Step 3 is called classification because it is our objective in the step, although it consists of several experiments using different core concepts for classification such as search, tagging, supervised classification using ML and classification using recommendations based approaches. The approaches use NLP concepts and state-of-the art ML and DL based models (base or fine-tuned). As not all of the experiments were the best classification approaches towards a working solution but were significant enough due to the knowledge gained to be able to reach the final approach of semantic textual similarity, they will be discussed in order.

5.1 EDI Parsing

The first step to be able to utilize the highly unstructured EDI data was parsing. Most of the cargo based rail, ports and airports in the world still use the file format EDI for their transactions. An EDI trade document can be defined as a data file formatted using one of several Electronic Data Interchange (EDI) standards. It contains trade or business data stored in a plain text format and is used for transferring that data between multiple organizations. The first goal was to look deeper and extract relevant trading information from the trade document most importantly the description of the content being shipped. A deeper part of parsing was to extract commodity classification codes if there was one within the description. The challenge was to handle the randomness of the descriptions from different shippers for different goods since no two files are exactly similar. The description often contains several numbers such as postal code, phone no, fax no, etc. So distinguishing between them to find out the commodity code also proved to be a challenge. All confidential and proprietary information was redacted before the data was provided for experiments reported in this thesis. Over 30,000 EDI trade documents were analyzed from different shippers. Each document containing from one to hundreds of trade transactions. To deal with this overwhelming amount of complex text data we tried a few different parsing approaches before deciding on a final one.

5.1.1 Existing Open-source EDI parsers

Even though there are only a few EDI parsers available out there, none of them are exclusively built to parse the trade data to the depth needed for research. We tried two most popular EDI parsers, the NPM parser and Caliber Health's EDI parser [82]. Unlike XML, EDI uses almost same set of tags for different standards. Since these parsers were more for Healthcare purposes they still did not provide sufficiently accurate results on some of the trade standards such as EDI 310 and 311.

5.1.2 Enterprise Solutions

We looked towards trying some enterprise-level solutions for data cleansing such as Data Ladder and Openprise but they were unable to handle the randomness of the information contained within the EDI trade documents from different shippers [83, 84].

5.1.3 RegEX Parser

Finally, I decided to build my own EDI parser. I managed to combine a multitude of regular expression-based methods to create a rule-based parser that can address the variations of the EDI manifests from different shipping carriers. A set of another 107 regular expressions were needed to extract the harmonized commodity codes in the raw description if any existed. The reason for that was the different wording of commodity code by different shippers. For example, all these instances mean the same: 'H.S. CODE', 'Commodity Classification NO', 'NCM', 'Tariff NO', 'HTS', 'HC

CODE', 'CODE', 'HS' etc. Since the trade terminals such as airports, ports, railway junctions process more than thousands of shipments every day, this novel approach was designed with the intention to reduce manual human effort and error. This results in checking of the shipping containers in time and reduce various risks that we discussed before. However, this parser constantly needs to be updated if conflicts arise due to unseen data format. The RegEx parser worked much more efficiently with 87 percent accuracy and I was able to retrieve the set of information I needed to proceed to the next step.

5.2 Commodity Description Pre-processing

Next, I needed to clean the description part of the transactions and remove irrelevant information. Primarily I only needed the information that stated what was the commodity or the product that was being shipped. Several fields were discarded as not beneficial. So, first I implemented basic pre-processing such as removing noise, numbers, English stop words and then I did some high level pre-processing such as text normalization, spell correction, translation to English with Google translate API where applicable, removing repeated sentences and adding a new list of stop words constructed through repeated observations. This new list of stop words contained words that did not help in identifying the key commodity in the description. Some examples are generic words when identifying bulk amount of a product such as: 'container', 'bags', 'kilograms', 'net', 'lbs', etc.

Let us take a look at a sample example to better understand what information we want to parse and pre-process from the EDI trade documents. If we apply the steps of parsing on Fig 3.1 (sample EDI file) we extract the following description:

```
CARGO IS STOWED IN A REFRIGERATED CONTAINER SET AT THE
SHIPPER'S REQUESTED CARRYING TEMPERATURE OF -2 DEGREES
CELSIUS FREIGHT PREPAID WET SALTED SAITHE (POLLACHIUS
VIRENS) NET WEIGHT: 22.570,00 KGS HS-CODE: 03056980
```

and Commodity Code: "03056980". After pre-processing we get the description as:

```
wet salted saithe pollachius virens
```

For the next step of the approach, the above information is exactly what we need, only the product description and no other details about it. The example above is a very basic one for understanding, official every-day EDI trade data contains very tangled product descriptions mixed with addresses, numbers and more out of field information.

Since each transportation corporations aka shippers have their own way of writing product descriptions in EDI trade documents, I had to build different description cleaning script or a text-preprocessor for each of them. Out of all the available data I had from different shippers, I was able to work with trade data from four of them. They are mentioned as Shipper 1 to 4. For the rest of the shippers, the amount of data was very minimal.

5.3 HS Code Classification

In this section, I will discuss different approaches that have been explored towards the objective of correctly classifying commodity descriptions without a commodity code into correct commodity codes or editing existing incorrect codes.

5.3.1 Classification Using Named Entity Recognition

Just like any other text document, there are particular terms that represent specific entities that are more informative and have a unique context. For our project that information is the product name in the description of a trade document. As our very first approach with a beginner-level understanding of the entire problem as a whole, we tried named entity recognition (NER) to correct or insert commodity codes in trade documents. The plan was to use the 'PRO' tag in NER to identify product names in both trade document and the UN comtrade sheet, then we match the product names from both descriptions to find the commodity code from the UN comtrade sheet, since that is an index for all types of traded goods. To accomplish this task, first, we tried the in-built product tagger in Spacy. Next, we trained Spacy with product entities from the UN comtrade sheet.

5.3.2 Classification Using Machine Learning with Historical Trade Data

Next I focused on machine learning-based approaches to building a classifier using historical trade data. For this task, I needed a labeled data set. A labeled data set in this scenario are trade documents containing correct commodity codes in the product description section. So, to employ text classification, the pre-processed product description acts as the text and the commodity code within the description acts as the label. With the client's recommendation I went on to conduct experiments with data from a shipper they trust to make the lowest amount of mistakes when enclosing trade documents. I started the machine learning-based approach with non-neural network-based classifiers.

According to the working principles of machine learning algorithms, they can be divided into three categories: 1) Unsupervised Algorithms, 2) Supervised Algorithms, and 3) Semi-supervised algorithms. To address our research problems, the classifiers were implemented using only supervised machine learning approaches.

Using the supervised approach, the classifiers are trained with the text and their respective labels. These models learn with the existing data and learn until optimal performance is achieved on the training data. The classifiers are further evaluated on the unforeseen data.

5.3.3 Classification using Semantic Textual Similarity via Deep Learning

A primary ambition of this research is to help the cargo terminals such as rail, ports, airports to be able to automate their process of shipment validation and reduce risk. A lot of the operations done at most of these cargo terminals in the world are manual labor, including shipping container code assignment. Someone manually types in the commodity code for a transaction and when done in thousands every day the process is prone to human error which happens to be one of the major concerns for the ports. An example would be a transaction with an apple in the description with a commodity code that is of a bicycle. In this final step of the methodology, we deal with correcting shipping manifests with incorrect or no commodity codes. We take a pre-processed product description from a trade manifest and match it semantically against all of the descriptions in the UN commodity reference sheet using sentence transformer models. We then try to suggest the top k matched commodities based on the cosine similarity score. Due to the sensitivity of the information, all recommended changes should be validated by a human prior to changes being made in the trade document. For the task of semantic textual similarity we used state of the art, pre-trained sentence transformer models to date: Google's Universal Sentence Encoder, Infersent 1 and 2 from Facebook research [5], Sentence Bert [7] and its base and large versions of RoBERTa, XLM-R and DistilBERT.

A schematic diagram of our STS approach is shown in Figure 5.1. Document preparation has two steps. First, parsing of trade documents is performed as per Section 5.1. Second, we pre-process the trade document description and UN commodity descriptions as per Section 5.2. The pre-processed commodity reference descriptions become our corpus while the trade document description acts as a query. Next, we use our STS models to generate query and corpus embeddings. We then compare each corpus embedding against the query embedding based on cosine similarity. Top k commodities are then recommended from the corpus with their respective HS codes based on the cosine similarity score. The human expert picks the correct choice from the recommendations and proceeds to the next.

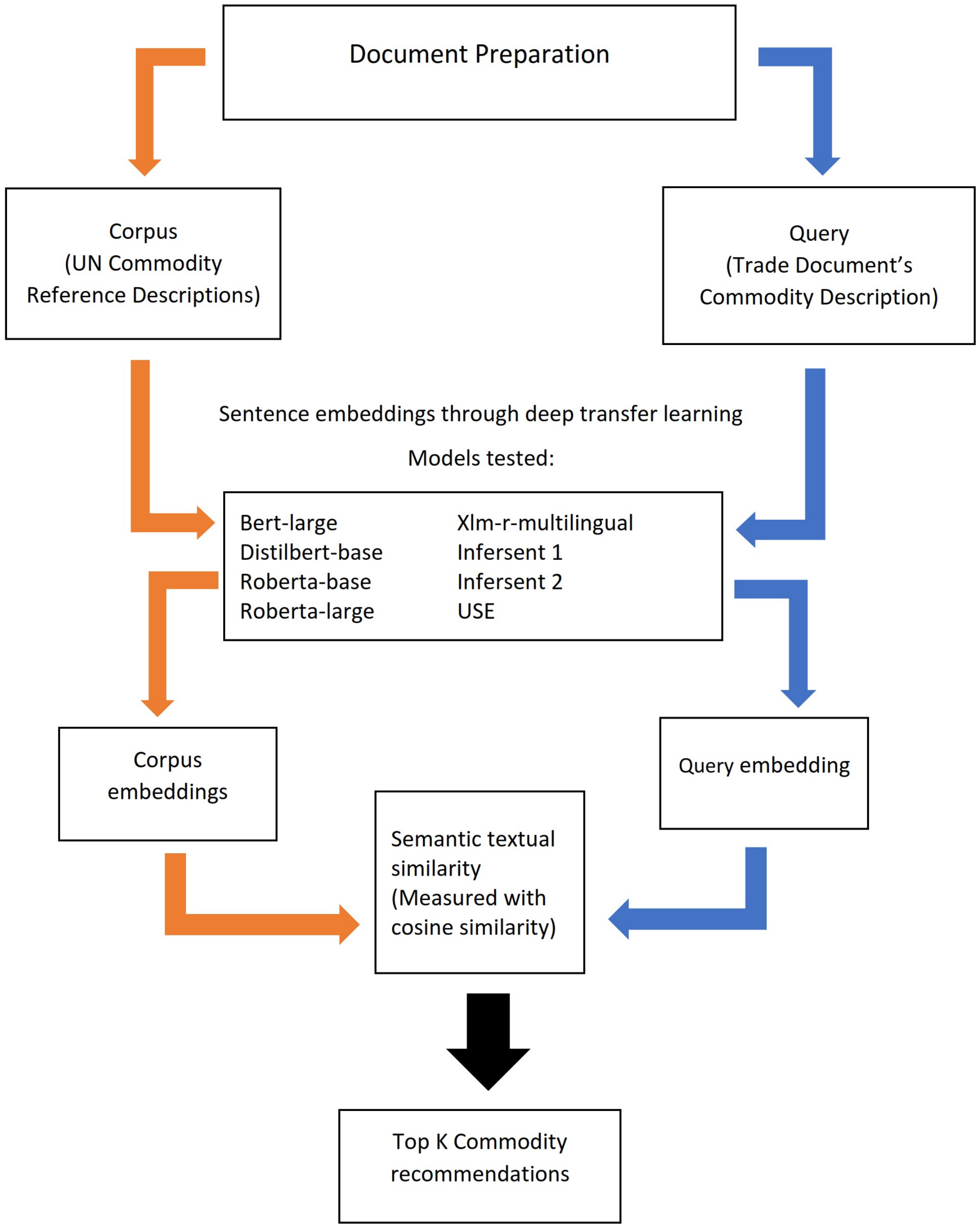

Document Preparation
Corpus (UN Commodity Reference Descriptions)
Query (Trade Document's Commodity Description)
Sentence embeddings through deep transfer learning
Models tested:
Bert-large
Distilbert-base
Roberta-base
Roberta-large
Xlm-r-multilingual
Infersent 1
Infersent 2
USE
Corpus embeddings
Query embedding
Semantic textual similarity (Measured with cosine similarity)
Top K Commodity recommendations

Figure 5.1: Schematic diagram of classification using Semantic Textual Similarity

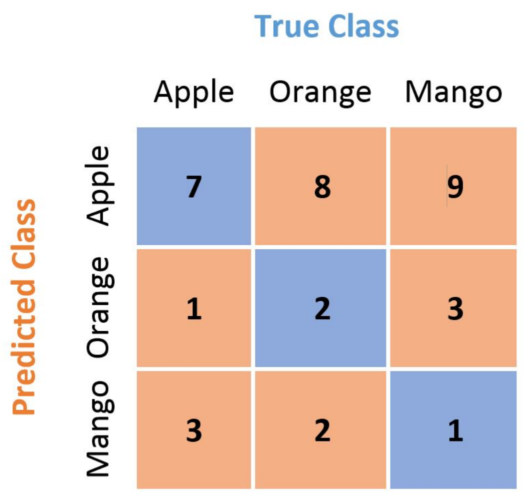
Chapter 6

Results and Evaluation

In this section, I will present and analyze the results from Named Entity Recognition approach and classification approach using supervised machine learning algorithms. For classification using supervised ML algorithms, we have used the Term Frequency-Inverse Document Frequency (TF-IDF) Model for text features extraction technique. To evaluate the performance of the NER and Classification approach using ML training, we use precision, recall and F1-score that are in general used for binary text classification. For classification using Semantic Textual Similarity (STS) approach, we define our own evaluation criteria based on Precision at k due to specific needs of the application task.

6.1 Evaluation Matrix

In this section, the methods to evaluate the performance of the classifiers are discussed. One of the most efficient tabular visualization models to represent the performance of a classifier is a confusion matrix. A confusion matrix shows the combination of the actual and predicted classes. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class. It is a good measure for multi-class classification such as ours, to account for whether models can account for the overlap in class properties and understand which classes are most easily confused. A sample confusion matrix for multi-class classification is shown in Figure 6.1.



Predicted Class \ True Class	Apple	Orange	Mango
Apple	7	8	9
Orange	1	2	3
Mango	3	2	1

Figure 6.1: Confusion Matrix for Multi-Class Classification [8]

For a binary classification, there exist only two classes to classify, a positive and a negative class. So, for such classification it is easier to define and calculate True Positives, False Positives, True Negatives and False Negatives:

True Positive (TP): It refers to the number of predictions where the classifier correctly predicts the positive class as positive.

True Negative (TN): It refers to the number of predictions where the classifier correctly predicts the negative class as negative.

False Positive (FP): It refers to the number of predictions where the classifier incorrectly predicts the negative class as positive.

False Negative (FN): It refers to the number of predictions where the classifier incorrectly predicts the positive class as negative.

However, in our experiments of multi-class classification, there are no positive or negative classes here. So, we find TP, TN, FP and FN for each individual class. Let's assume we have three classes, Apple, Orange and Mango. Let's take a look at a sigle class Apple and calculate it's TP, TN, FP and FN values from the confusion matrix in Figure 6.1.

$$TP = 7, \quad TN = (2 + 3 + 2 + 1) = 8$$

$$FP = (8 + 9) = 17, \quad FN = (1 + 3) = 4$$

I found it efficient to use confusion matrix as evaluation criteria for the machine learning models as it gives very simple, yet efficient performance measures for the models. Here are the performance measures I have used from the confusion matrix.

Precision: Precision can be defined as the number of true positives divided by the summation of true positives and false positives. Precision is a very useful metric as it measures the correctly classified instances against all the original instances of that specific class. Precision can be formulated as:

$$Precision = \frac{TP}{TP + FP}$$

Recall: Recall can be defined as number of true positives divided by summation of true positives and false negatives. Recall can be formulated as:

$$Recall = \frac{TP}{TP + FN}$$

F1-score: It combines precision and recall into a single measure. Mathematically it is the harmonic mean of precision and recall. It can be calculated as follows:

$$F1\text{-}Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

6.2 Hardware Requirements and Configuration

Most of the experiments were executed on a high-performing personal PC, however, some testing scripts were executed using Google Colab which is a platform that allows us to write and execute Python scripts in the browser and it does not require the installation of dependencies separately. To calculate and compare the training or execution time of the classifiers, I used a personal computer that has the following configuration. In the personal computer, all of the experiments were performed using Python installed on an Anaconda Distribution using TensorFlow-GPU 2.0 support and NVIDIA CUDA 10.0 with cuDNN.

- Operating System: Microsoft Windows 10 Home 64-bit Operating system, x-64 based processor
- Version: 10.0.18363 Build 18363

- Processor: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
- RAM: 16.0 GB
- GPU: NVIDIA GeForce RTX 2070 with Max-Q Design

6.3 Classification Using Named Entity Recognition Approach

NER was my first approach towards solving the problem of HS code classification. However, both SpaCy's built-in product entity tagger and trained entity tagger using UN commodity names from the UN trade sheet (see section 2.1.3) turned up with really poor results. The performance results are shown in Table 6.1.

Precision	Recall	F1-score
14.3214	27.2186	18.7678

Table 6.1: SpaCy NER results (trained) on shipper 1 data

One of the primary reasons for low evaluation scores is the difference between the shipper's way of identifying a product name and the UN comtrade sheet's description for the same product (as discussed in Chapter 1 – Terminological gaps). Shippers would like to enclose their product as 'MP3 player' but according to the trade index by the UN it belongs to 'Sound recording or reproducing apparatus'.

6.4 Classification using Machine Learning with Historical Trade Data for Training Approach

In this section, we present results of using the common supervised machine learning algorithms. Although the odds were high, these methods were executed to test if they work at all even for a low number of instances. We received poor evaluation results which included a large number of misclassifications. The reason for such results can be a few important things. First, due to the difference in data type, we test our algorithms on different shippers. Our best data was shipper 1 which had 9684 labeled instances and 9233 without labels. To improve our results we optimized our models using cross-validation and hyperparameter tuning using methods such as grid search. However, we have 6198 classes. Therefore, the number of classes are far too many in comparison with the amount of available data. The number of repeated classes in the dataset are also very few thus resulting in far too many outliers. From the evaluations, even if SVM got some of the classifications right, the importers can not really afford misclassifications. Since in this method the classifier assigns a label to every non-labeled instance in the test data, it is again costly to detect the errors and revert back once a mistake has been made. This approach is not sufficient especially when we are trying to minimize error, lower manual human effort and the time it takes to process the trade documents. Another reason was confidence in classification, due to the sensitivity of the data we could not just confidently assign a class from looking at accuracy or F1-Score. Due to the amout of limited data we did not have a good measure to confidently assign a classification score or a general confidence score in this process. So, we could not afford to go with this approach. There are opportunities of increasing the classifier accuracy over time with more accurate training data, however, there's still doubt about how well it is going to work for more than six thousand classes. The next method relies on a text similarity approaches where we hope that we can address some of the misclassifications in the supervised methods.

	Precision	Recall	F1-score
Multinomial NB	31.4599	46.2208	37.4379
Random Forest	27.5685	34.1327	30.5014
SVM	37.2564	48.4151	42.1090

Table 6.2: Classification results using ML algorithms on shipper 1 data (trained)

6.5 Classification Using Semantic Textual Similarity (STS) Approach

To correctly identify the usefulness of our semantic-based approach, calculating simple accuracy or precision and recall that we use for binary or multi-class classifications was not enough. Since we are trying to suggest top k numbers of relevant commodity matches based on similarity scores, the process is almost identical to a search engine or a recommender system. To evaluate such systems the most common approach is to use precision at k or recall at k. In an information retrieval system that retrieves a ranked list, the top-k documents are the first k in the ranking. The previous supervised methods could also be made to rank the likely classes but it is a less transparent process whereas the STS approach naturally introduces a similarity measure as a way of ranking most likely classes and it is convenient and also motivated by similar real-world applications to introduce these measures.

We define these measures in the next subsection.

6.5.1 Precision at k

Precision at k is the proportion of the top-k documents that are relevant.

Definition 6.5.1 (Precision at k)**.** If r relevant documents have been retrieved at rank k (among the first k documents), then

$$Precision\ at\ k = \frac{r}{k}$$

In the above definition, ranked documents are defined as the top k documents that are displayed as results by the recommender system. Relevant documents are the expected classes that should appear as relevant in the top k result.

While this definition is accurate, for the problem we have at hand this measurement is simply not good enough in terms of calculating precision. When we are trying to identify the efficiency of our semantic models, we test them on labelled dataset. Which means that we know the class (HS code) to expect in the top k recommendations. So, what we are most interested in, is to know if this correct commodity code appeared in the top k suggestions by the model. Similarly, a relevant commodity choice in this regard would be the commodity with HS code belonging to the same code group as the correct commodity (detailed explanation with example in Section 6.5.3). When we are looking to be precise, that is the information we care about specific to this problem. So, for our scenario we define a new evaluation criteria called recommendation precision.

Definition 6.5.2 (Recommendation Precision at k)**.** If r relevant documents have been retrieved at rank k, then recommendation precision at k is 1, if the correct commodity code with commodity description i appears within rank k.

6.5.2 Recall at k

Recall gives a measure of how accurately our model is able to identify the relevant data.

Definition 6.5.3 (Recall at k)**.** If r relevant choices have been retrieved at rank k and total number of relevant choices is n, then

$$Recall\ at\ k = \frac{\text{number of relevant choices at rank } k}{\text{total number of relevant choices}} = \frac{r}{n}$$

6.5.3 Example Evaluation of Recommendation Precision and Recall at k

For clarity let us look at an example evaluation. Note that, as mentioned before, our approach is similar to the approach a search engine or recommender system takes but is not exactly the same when it comes to evaluation. We do not really want to rank the most relevant results, our intention is to have the correct commodity and commodities within the same code group appear in top k results. We care about at which position out of top k we are getting the correct commodity choice. The intention is to help the appropriate authority to be able to easily correct or assign a commodity code to a trade document where there exists a commodity description without a commodity code or an incorrect one. Let us take a look at an example trade manifest description:

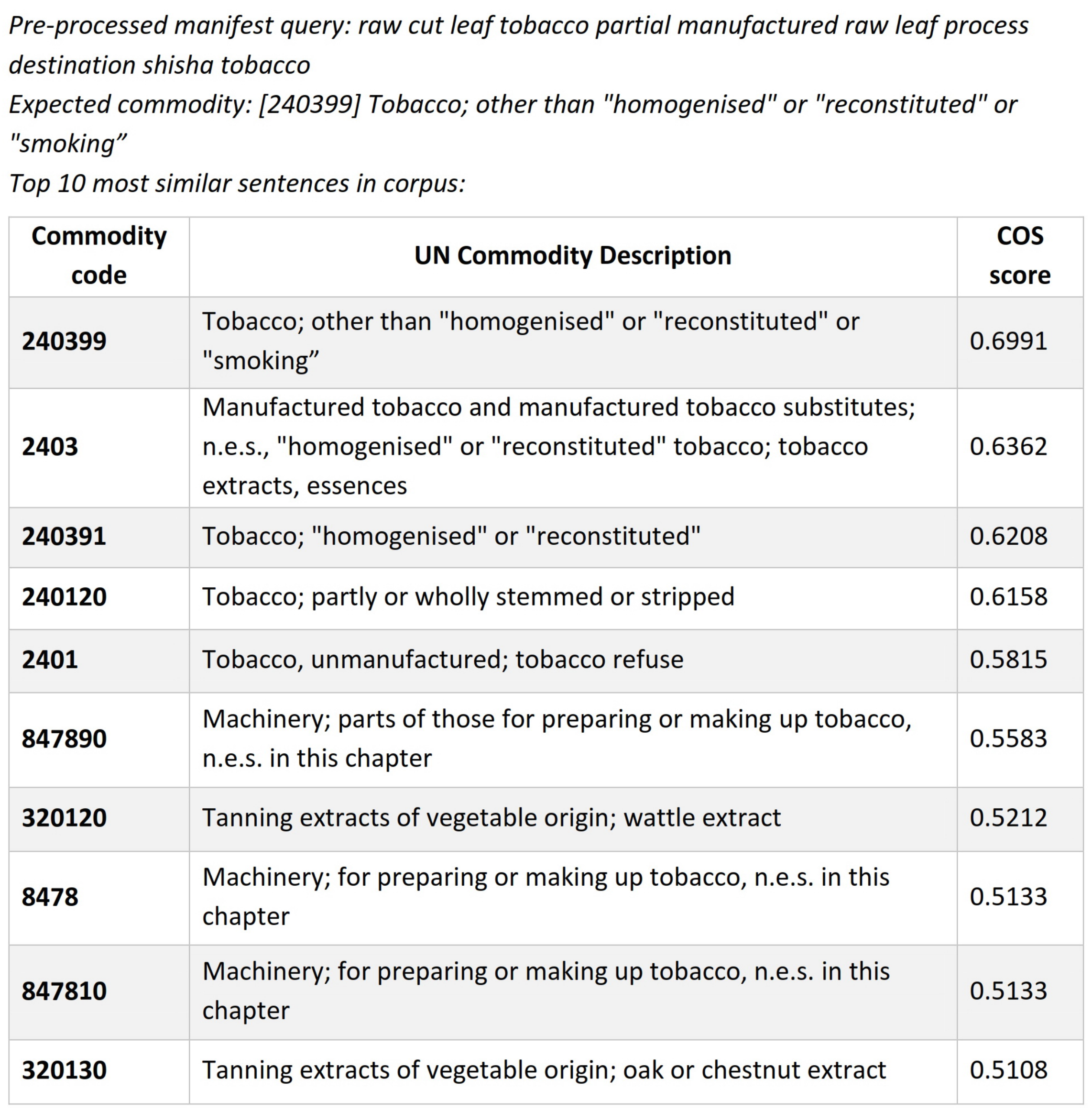
```
114 BOXES RAW CUT LEAF TOBACCO PARTIAL MANUFACTURED RAW
LEAF FOR FURTHER PROCESS AT THE DESTINATION FOR SHISHA
TOBACCO PRODCUT INV NO: XX/XXX/X DT: XX/XX/XXXX CODE NO:
240399 P.O.NO:XXX ...
```

Here, we first use parsing to retrieve the product description and commodity code which is 240399. We use the commodity code to look up the desired commodity description in the UN provided commodity trade sheet in Table 6.3. In this process we also find out that there are four different commodities within the 2403 HS code group.

Classification	Code	Description	Code Parent	Level	isLeaf
H0	2309	Preparations of a kind used in animal feeding	23	4	0
H0	230910	Dog or cat food; put up for retail sale, used in animal feeding	2309	6	1
H0	230910	Dog or cat food; (not put up for retail sale), used in animal feeding	2309	6	1
H0	24	Tobacco and manufactured tobacco substitutes	TOTAL	2	0
H0	2401	Tobacco, unmanufactured; tobacco refuse	24	4	0
H0	240110	Tobacco, (not stemmed or stripped)	2401	6	1
H0	240120	Tobacco; partly or wholly stemmed or stripped	2401	6	1
H0	240130	Tobacco refuse	2401	6	1
H0	2402	Cigars, cheroots, cigarillos and cigarettes; of tobacco or of tobacco substitutes	24	4	0
H0	240210	Cigars, cheroots and cigarillos; containing tobacco including the weight of every band, wrapper or attachment thereto	2402	6	1
H0	240220	Cigarettes; containing tobacco	2402	6	1
H0	240290	Cigars, cigarillos and cheroots; containing tobacco substitutes including the weight of every band, wrapper or attachment thereto	2402	6	1
H0	2403	Manufactured tobacco and manufactured tobacco substitutes; n.e.s., “homogenised” or “reconstituted” tobacco; tobacco extracts, essences	24	4	0
H0	240310	Tobacco, smoking; whether or not containing tobacco substitutes in any proportion	2403	6	1
H0	240391	Tobacco; “homogenised” or “reconstituted”	2403	6	1
H0	**240399**	**Tobacco; other than “homogenised” or “reconstituted” or ”smoking”**	**2403**	**6**	**1**
H0	25	Salt; sulphur; earths, stone; plastering materials, lime and cement	TOTAL	2	0
H0	2501	Salt (including table salt and denatured salt); pure sodium chloride whether or not in aqueous solution; sea water	25	4	0

Table 6.3: A section of UN approved commodities trade sheet

Next, we evaluate our results in terms of recommendation precision and recall. We take a look at the recommendation results for our commodity description (defined as Pre-processed manifest query) in Figure 6.2.



Pre-processed manifest query: raw cut leaf tobacco partial manufactured raw leaf process destination shisha tobacco
Expected commodity: [240399] Tobacco; other than "homogenised" or "reconstituted" or "smoking”
Top 10 most similar sentences in corpus:

Commodity code	**UN Commodity Description**	**COS score**
240399	Tobacco; other than "homogenised" or "reconstituted" or "smoking”	0.6991
2403	Manufactured tobacco and manufactured tobacco substitutes; n.e.s., "homogenised" or "reconstituted" tobacco; tobacco extracts, essences	0.6362
240391	Tobacco; "homogenised" or "reconstituted"	0.6208
240120	Tobacco; partly or wholly stemmed or stripped	0.6158
2401	Tobacco, unmanufactured; tobacco refuse	0.5815
847890	Machinery; parts of those for preparing or making up tobacco, n.e.s. in this chapter	0.5583
320120	Tanning extracts of vegetable origin; wattle extract	0.5212
8478	Machinery; for preparing or making up tobacco, n.e.s. in this chapter	0.5133
847810	Machinery; for preparing or making up tobacco, n.e.s. in this chapter	0.5133
320130	Tanning extracts of vegetable origin; oak or chestnut extract	0.5108

Figure 6.2: Universal Sentence Encoder results for semantic textual similarity

Based on the needs of the practical application we took the value of $k = 10$. We notice that the desired commodity with HS Code **240399** appears in the very first result out of ten recommendations which is ideally what we want. We also observe the other commodity recommendations with very similar meanings but different commodity code initials. From a product perspective and conditions based on practical applications, the calculation of precision at 10 is not enough for evaluation since we have semantically similar yet different commodities with different HS commodity codes. So, instead, we set recommendation precision for this particular result to 1. Then we calculate the average recommendation precision for all of the test data sets.

To calculate recall at 10, we calculate how many of the relevant commodities appeared in our result. We observe, not all of the relevant commodities starting with 2403 code group (Table 6.3) appears in the results (Figure 6.2). We notice only 3 relevant commodities from the 2403 code group out of a total of 4 appeared in our recommendation results. So for this instance,

$$\textit{Recall at 10} = \frac{\text{number of relevant choices at rank k}}{\text{total number of relevant choices}} = \frac{3}{4} = 0.75$$

When computing recall at k, in rare instances we might face a situation when the total number of relevant items is zero. In that case, we set recall at k to be 1. We cannot compute recall at k since we cannot divide by zero. This makes sense because we do not have any relevant item that is not identified in our top k results.

6.5.4 Model Comparisons

Now that we know how to evaluate the models used, it is time to compare and pick the best one. To recap, we want high recommendation precision and high recall but we also care for our desired commodity to appear in the top-most results out of the top k. We also compare the STS benchmark score for individual models. STS Benchmark comprises a selection of the English data sets used in the STS (semantic textual similarity) tasks in order to provide a standard benchmark to compare among meaning representation systems.

Results for Shipper 1 (total transactions = 9684):

	Model	**Recommendation Precision at 10**	**Recall at 10**	**STSb Performance**
Bert	stsb-bert-large	51.2701	75.9963	85.29
	stsb-distilbert-base	51.0016	71.7678	85.16
	paraphrase-distilroberta-base-v1	41.4807	64.8868	81.81
	stsb-roberta-base	44.7542	68.6843	85.44
	stsb-roberta-large	48.2755	71.8032	86.39
	stsb-xlm-r-multilingual	51.4766	77.2593	83.50
Infersent	Infersent 1 (glove.840B.300d)	39.2399	62.6962	75.5
	Infersent 2 (fast text crawl-300d-2M-subword)	25.4646	46.0977	-
	Infersent 2 (fast text crawl-300d-2M)	31.77	58.5228	78.4
USE	Universal Sentence Encoder Large v5	53.1495	75.2437	74.92

Table 6.4: Comparison of the models used based on recommendation precision and recall for shipper 1 data

For shipper 1, we can observe that four of the models are performing at a recommendation precision greater than 50 and recall greater than 70 (Table 6.4). Note that we do not calculate F1 score since recommendation precision and precision are not the same. To compare among these top models now we take a look at their recommendation distributions. The preferred one would be the one to have the desired commodity appear in its top results maximum number of times. For shipper 1 we present those results in Figure 6.3. For shipper 2, 3 and 4 we present the results respectively in Tables 6.5, 6.6, and 6.7; and Figures 6.4, 6.5, and 6.6.

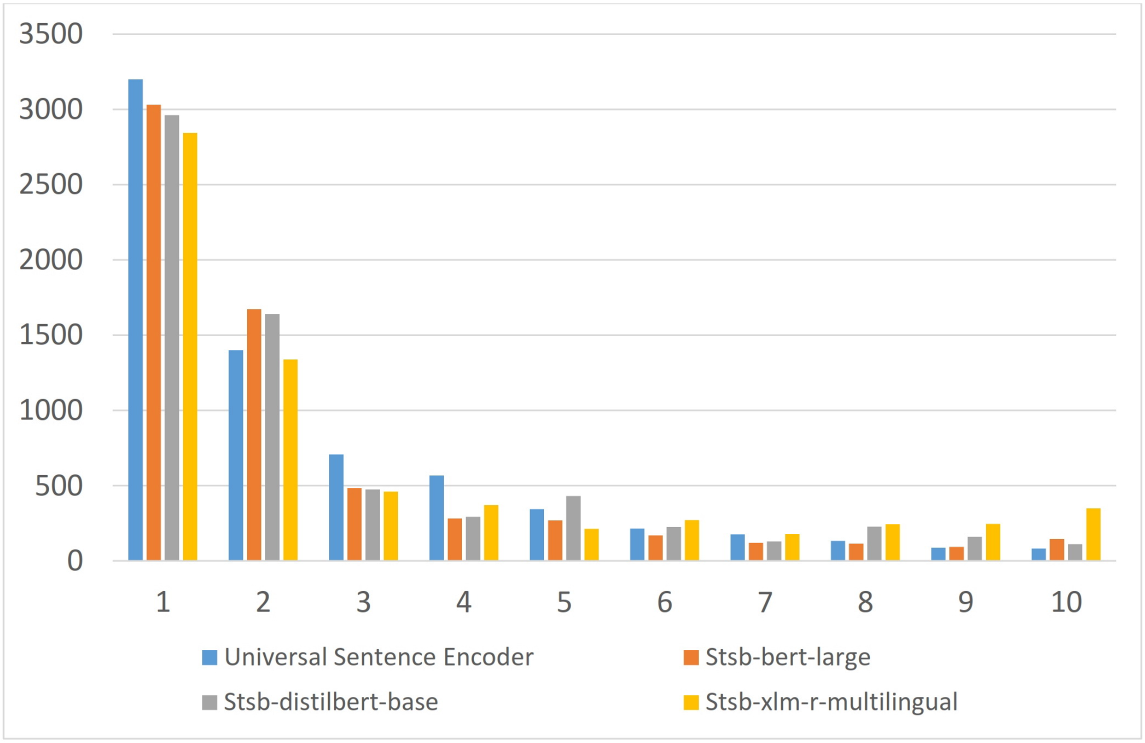


Figure 6.3: Comparison of models used based on recommended choices for shipper 1 data

Results for Shipper 2 (total transactions = 6823):

	Model	**Recommendation Precision at 10**	**Recall at 10**	**STSb Performance**
Bert	stsb-bert-large	64.3121	75.0320	85.29
	stsb-distilbert-base	66.1487	77.3122	85.16
	paraphrase-distilroberta-base-v1	39.9179	47.6408	81.81
	stsb-roberta-base	52.0651	66.7329	85.44
	stsb-roberta-large	66.7305	78.1038	86.39
	stsb-xlm-r-multilingual	68.3380	80.3255	83.50
Infersent	Infersent 1 (glove.840B.300d)	42.2399	59.7216	75.5
	Infersent 2 (fast text crawl-300d-2M-subword)	27.0012	48.0324	-
	Infersent 2(fast text crawl-300d-2M)	41.2130	58.9861	78.4
USE	Universal Sentence Encoder Large v5	70.0971	81.3562	74.92

Table 6.5: Comparison of the models used based on recommendation precision and recall for shipper 2 data

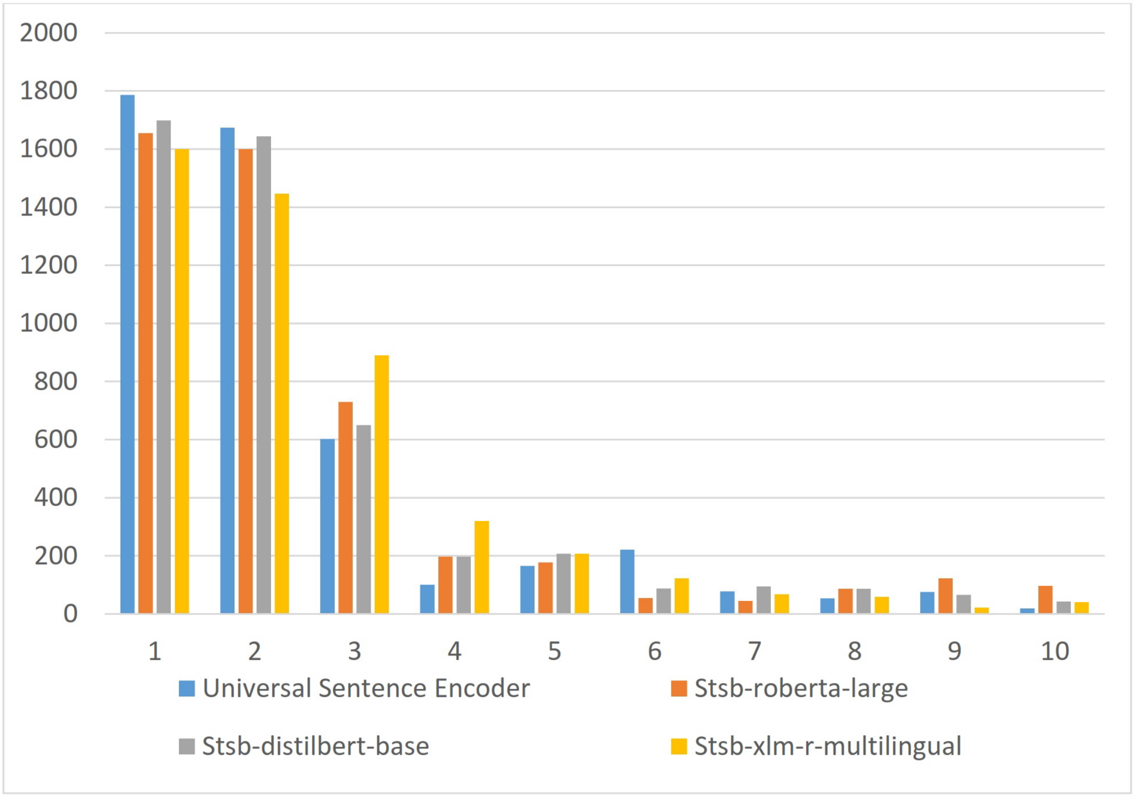


Figure 6.4: Comparison of models used based on recommended choices for shipper 2 data

Results for Shipper 3 (total transactions = 12248):

	Model	Recommendation Precision at 10	Recall at 10	STSb Performance
Bert	stsb-bert-large	42.4574	59.7633	85.29
	stsb-distilbert-base	43.6790	62.0234	85.16
	paraphrase-distilroberta-base-v1	27.7547	39.2123	81.81
	stsb-roberta-base	41.9041	66.7329	85.44
	stsb-roberta-large	45.7305	65.0413	86.39
	stsb-xlm-r-multilingual	50.1117	71.0478	83.50
Infersent	Infersent 1 (glove.840B.300d)	30.4541	42.0151	75.5
	Infersent 2 (fast text crawl-300d-2M-subword)	18.8562	35.3203	-
	Infersent 2 (fast text crawl-300d-2M)	28.2216	34.7421	78.4
USE	Universal Sentence Encoder Large v5	43.9430	68.4054	74.92

Table 6.6: Comparison of the models used based on recommendation precision and recall for shipper 3 data

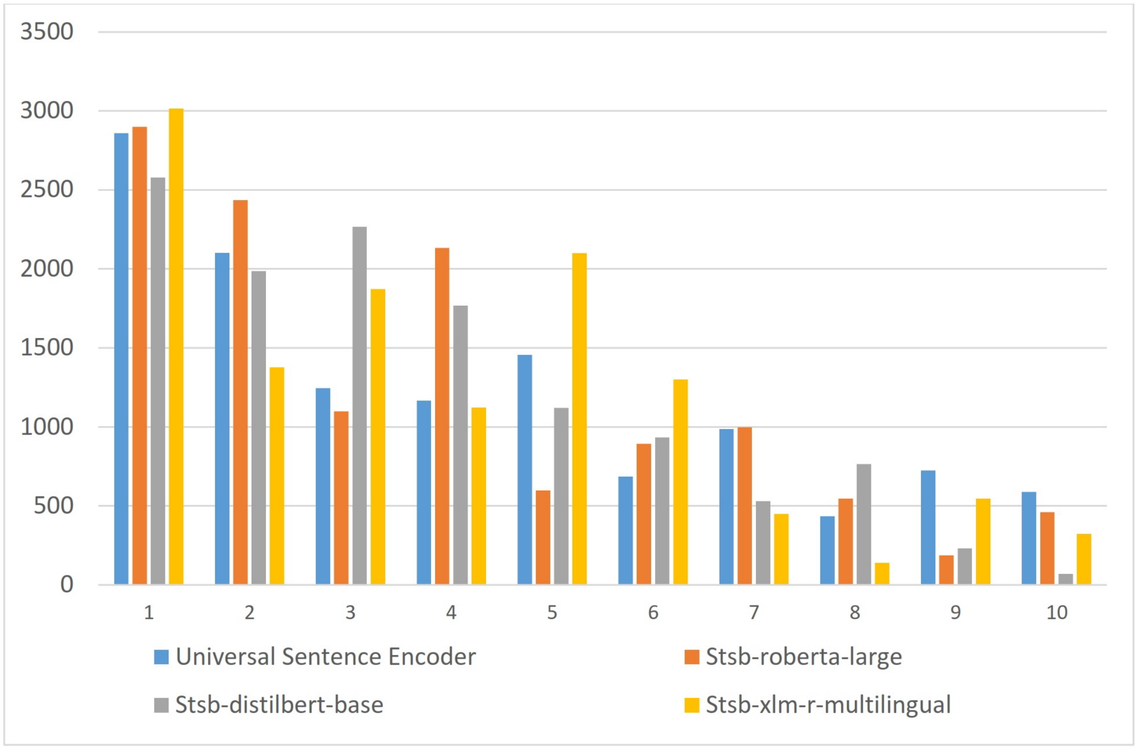


Figure 6.5: Comparison of models used based on recommended choices for shipper 3 data

Results for Shipper 4 (total transactions = 4264):

	Model	**Recom-mendation Precision at 10**	**Recall at 10**	**STSb Performance**
Bert	stsb-bert-large	55.7634	73.2344	85.29
	stsb-distilbert-base	56.6790	71.0234	85.16
	paraphrase-distilroberta-base-v1	42.7851	63.3523	81.81
	stsb-roberta-base	45.7896	66.1312	85.44
	stsb-roberta-large	56.1433	69.2319	86.39
	stsb-xlm-r-multilingual	55.8482	72.4667	83.50
Infer-sent	Infersent 1 (glove.840B.300d)	40.5390	60.7113	75.5
	Infersent 2 (fast text crawl-300d-2M-subword)	26.0016	47.5331	-
	Infersent 2 (fast text crawl-300d-2M)	38.1114	54.0466	78.4
USE	Universal Sentence Encoder Large v5	58.3112	73.6810	74.92

Table 6.7: Comparison of the models used based on recommendation precision and recall for shipper 4 data

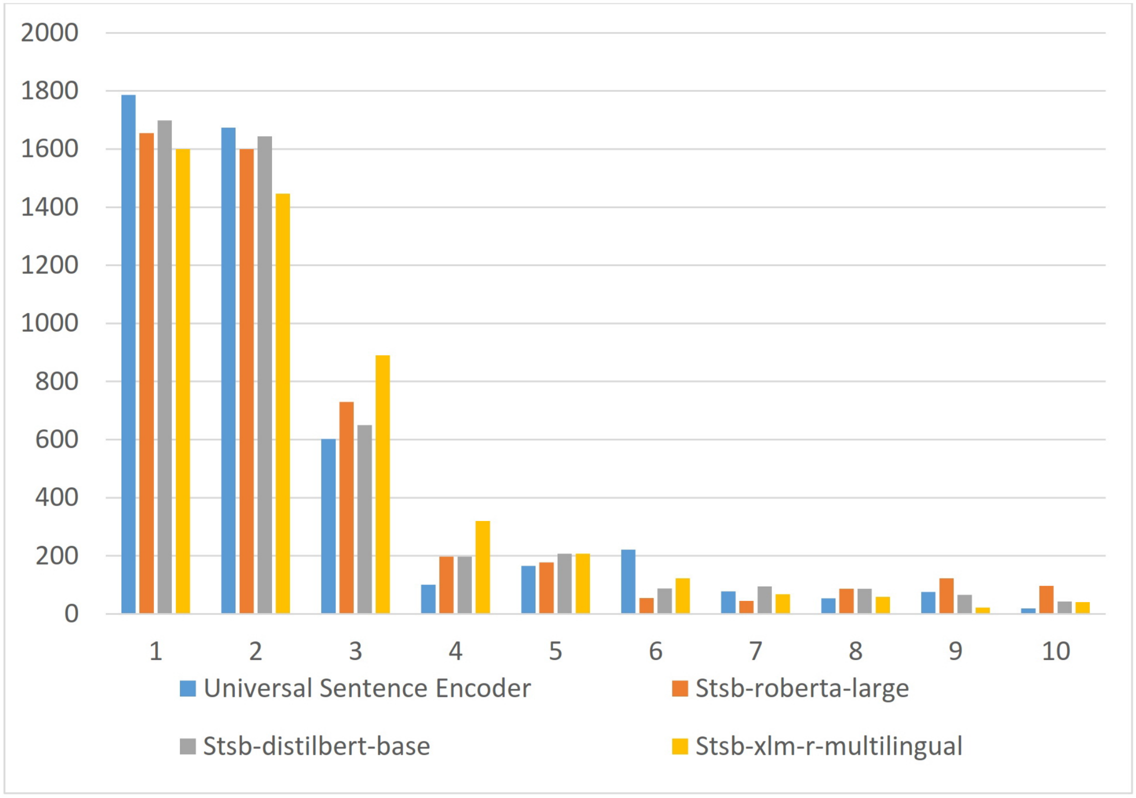


Figure 6.6: Comparison of models used based on recommended choices for shipper 4 data

Now to compare our models and pick the best one for the data from four primary shippers, we first observe the top four performing models for **Shipper 1** (Table 6.4):

1. Sentence BERT-DistilBERT-Base with recommendation precision and recall at 10 as: 51.0016 and 71.7678

2. Sentence BERT-BERT-Large with recommendation precision and recall at 10 as: 51.2701 and 75.9963

3. Sentence BERT-XLM-R with recommendation precision and recall at 10 as: 51.4766 and 77.2593

4. Google's USE with recommendation precision and recall at 10 as: 53.1495 and 75.2473

So, we can see for Shipper 1, BERT Large, XLM-R and USE are performing very close to each other while DistilBERT-BASE falls a bit behind on recall at 10.

The purpose of a recommender system is to suggest relevant items to users. Even though we prefer highest recommendation precision for our desired model, due to our approach being close to a recommender system we also want the best recall at 10. However, we cannot afford to pick the best model only on that criteria. Even if the system works in a human-assisted way, due to our primary task being HS commodity code classification, we care about which model shows the desired commodity on its top-most recommendation result. So, for Shipper 1 we take a look at Figure 6.3. and we observe Google's Sentence Encoder outperforming every other model by far on its first recommendation. So, even with a lower STS benchmark score, for Shipper 1 data we see USE as the top-performing model.

Next, we observe the top four performers for **Shipper 2** (Table 6.5):

1. Sentence BERT-DistilBERT-Base with recommendation precision and recall at 10 as: 66.1487 and 77.3122
2. Sentence BERT-RoBERTa-Large with recommendation precision and recall at 10 as: 66.7305 and 78.1038
3. Sentence BERT-XLM-R with recommendation precision and recall at 10 as: 68.3380 and 80.3255
4. Google's USE with recommendation precision and recall at 10 as: 70.0971 and 81.3562

So, we can see for Shipper 2, Google's USE performs with higher precision and recall at 10 values. XLM-R also performing almost at similar values but looking at Figure 6.4 we can see USE performing best at recommending the majority of its commodity choices at the top 2 ranks and XLM-R falling quite behind. Therefore, for Shipper 2 we choose Google's USE as the top performer.

Next, we observe the top four performers for **Shipper 3** (Table 6.6):

1. Sentence BERT-DistilBERT-Base with recommendation precision and recall at 10 as: 43.6790 and 62.0234
2. Google's USE with recommendation precision and recall at 10 as: 43.9430 and 68.4054

3. Sentence BERT-RoBERTa-Large with recommendation precision and recall at 10 as: 45.7305 and 65.0413

4. Sentence BERT-XLM-R with recommendation precision and recall at 10 as: 50.1117 and 71.0478

So, we can see for shipper 3, XLM-R outperforms every other model in terms of both recommendation precision and recall. We also observe XLM-R having a lead in recommending most of the desired commodities in its first choice (Figure 6.5). Now, to understand why XLM-R outperforms every other model by far in terms of recommendation precision we delve deeper. Sentence BERT's XLM-R is a multilingual supported model. Conveniently, Shipper 3 is a shipper from a foreign country where the language is significantly different from English. While most of the dataset is in English due to the trade products being imported in Canada, there are certain terms in the commodities which are in this foreign language. Therefore, XLM-R with its multi-language support performs better on foreign languages since every other model is trained on English datasets. So, for Shipper 3 with multi-language data, we choose XLM-R as the top performer.

Finally, we observe the top four performers for **Shipper 4** (Table 6.7):

1. Sentence BERT-XLM-R with recommendation precision and recall at 10 as: 55.8482 and 72.4667

2. Sentence BERT-DistilBERT-Base with recommendation precision and recall at 10 as: 56.6790 and 71.0234

3. Sentence BERT-RoBERTa-Large with recommendation precision and recall at 10 as: 56.1433 and 69.2319

4. Google's USE with recommendation precision and recall at 10 as: 58.3112 and 73.6810

We see for shipper 4, Google's USE outperforms every other model in terms of recommendation precision and recall at 10. USE also does better in recommending most of the desired commodities in its first two choices (Figure 6.6). Therefore, for Shipper 4 we choose USE as the top performer.

6.6 Comparison With Most Recent Literature

As previously discussed in Section 2.2.5, even after conclusion of the research project we tried to compare our approach with the recent HScodeNet in limited capacity [25]. Since, HScodeNet is hosted as a web application in Chinese language (Mandarin), we had to convert our trade document descriptions from English to Mandarin to be able to compare. However, we could not rely on the online translator (Google translate) for accurate translation of longer text descriptions so we had to resort to trying short text descriptions (max 3–4 words) manually one by one and compare the results. We show a few examples in Table 6.8.

Commodity: ‘Frozen lobster’, Mandarin translation: ‘冷冻龙虾’			
	Recommendation Precision at 10	Precision at 10	Recall at 10
HScodeNet	1 (3rd choice)	$\frac{2}{10} = 0.2$	$\frac{2}{10} = 0.2$
USE	1 (1st choice)	$\frac{9}{10} = 0.9$	$\frac{9}{10} = 0.9$
Commodity: ‘Pet foods’, Mandarin translation: ‘宠物食品’			
	Recommendation Precision at 4	Precision at 4	Recall at 4
HScodeNet	1 (1st choice)	$\frac{2}{4} = 0.5$	$\frac{2}{3} = 0.66$
USE	1 (2nd choice)	$\frac{3}{4} = 0.75$	$\frac{3}{3} = 1$
Commodity: ‘Oranges’, Mandarin translation: ‘橙子’			
	Recommendation Precision at 10	Precision at 10	Recall at 10
HScodeNet	0 (no results)	-	-
USE	1 (1st choice)	$\frac{5}{10} = 0.5$	$\frac{5}{5} = 1$
Commodity: ‘Fishing nets’, Mandarin translation: ‘渔网’			
	Recommendation Precision at 10	Precision at 10	Recall at 10
HScodeNet	1 (4th choice)	$\frac{1}{10} = 0.1$	$\frac{1}{3} = 0.33$
USE	1 (1st choice)	$\frac{3}{10} = 0.3$	$\frac{3}{3} = 1$

Table 6.8: Comparison of different commodity descriptions with HScodeNet

Let us take a look at the first example of ‘Frozen lobster’ which has a HS code 30612 (Table 6.8). For this commodity description, we find that the expected choice appears as third on HScodeNet while first for USE. We also find that, only two relevant commodities out of a total ten from the 306 HS code group appear in the suggestions given by HScodeNet, thus resulting in lower precision at 10 and recall at 10 values. It is interesting to see that quite a few of the top 10 choices given by HScodeNet are not relevant. The second commodity choice is handbags, fourth commodity choice is books, fifth commodity choice is shirts and so on. In our case, USE displayed nine relevant commodity suggestions from the same HS code group out of ten.

For the second example of Pet foods, we see that HScodeNet beats USE in providing the correct commodity in its first choice, however again falls a bit short on precision at 4 and recall at 4 values. We have used the value of k as 4 because HScodeNet only provides 4 results for the given query in Mandarin. Also, to note that after the first choice we get the second relevant choice from HScodeNet on choice number four. The second and third choices for Pet foods are books and plastic products which is completely irrelevant.

Next, if we take a look at the third example of oranges we observe that HScodeNet provides no results for such commodity. We have also tried other synonyms for oranges such as 'mandarin' but did not get any results.

Similarly, we have compared several other short description results with HScodeNet and found our approach to be superior for our given data. This comparison, however, is very limited since HScodeNet utilizes hierarchical and global spatial information in text and the description we have provided lacks such depth. Here is a sample description of Friction plate, as shown in the HScodeNet academic paper [25].

```
Friction plate | Can be used in various brands and models
of saloon cars, displacement: 1.4L to 2.0L | In the
automatic transmission of cars below 4th gear, clutch and
brake are used to fix any one or two of the three components
to achieve different transmission ratios | ......
```

The underlined text in the above description defines the hierarchical information HScodeNet uses to classify a commodity into an HS code category. This makes a full scale comparison extremely difficult since, the data we worked with lacks such structure. None of the commodity descriptions are even closer in terms of structure or hierarchy to what is shown above. The standard UN comtrade sheet we use to compare with trade document descriptions also lacks such structure and contains very short text descriptions for different commodities.

To explain further, HScodeNet is trained on the Chinese Customs data set, which means it can not classify a newer commodity it has not previously seen in the training data. This could be a reason for HScodeNet not displaying any results for oranges or mandarins. Our semantic based approach based on transfer learning performs better in such cases.

So, to summarize the comparison with HScodeNet, we can state that a full scale comparison was not possible due to the language barrier. However, for our data primarily available in English language with short commodity descriptions, the semantic based approach we proposed outperforms HScodeNet in terms of recommended choices and evaluation values of precision at 10 and recall at 10. For longer and structured commodity description texts such as the example of friction plate above, HScodeNet might perform better.

6.7 Discussion

In this chapter, we have presented and analyzed the research results in detail from all of the models used during experiments. We have compared the results and discovered that the semantic based approach is a superior one to take for the associated authority based on the data provided. From observations and model comparisons, we can also conclude that Google's Universal Sentence Encoder performs better than the other models for Shippers 1, 2 and 4. Facebook's XLM-R performs better for Shipper 3 with foreign language mixed data. We also compared our approach with the most recent and relevant research work in this domain: HScodeNet. While a full comparison was not possible due to reasons discussed in Sections 2.2.5 and 6.6, we concluded that for our dataset consisting of short commodity descriptions available primarily in English language, our approach is superior in performance and in minimization of risks. In the next section, we describe concluding remarks with limitations and possibility of future research directions to continue the work.

Chapter 7

Conclusion

We have attempted to address the challenging task of parsing EDI files to extract HS codes, cleaning the EDI description segment to extract the traded commodity description, and classifying HS a commodity code for the respective commodity description using deep transfer learning and obtained encouraging results. We also explored other methods such as named entities, elastic search, machine learning-based text classification techniques (Naïve Bayes, Random Forest, Support Vector Machine) resulting in poor results but ultimately evolving our approach to fill the observed shortcomings. As we saw in the discussion Section 6.7, we can conclude that for the current standard of the data from the four high-volume shippers, Google's Universal Sentence Encoder works the best for English-only data types while STSB-XLM-R-Multilingual works better for foreign language mixed data. It is interesting to note that even though the popular STS-benchmark has Google's Sentence Encoder performing at a lower benchmark compared to other STS models, it works surprisingly well in our unstructured and extremely noisy text data. The primary reason for USE to outperform other models is because it was trained on a variety of unstructured English data (web pages, articles, Wikipedia, discussion forums) while other models are trained on more structured data such as SNLI, CC Crawl and Books corpus. It's also important to note that a key difference in the performance of the models lies in the pre-processing of the commodity text description. We have obtained 40% improved relevant results over the default text pre-processors of the transformer models. Furthermore, our approach accounts for safety as of utmost importance using a human expert to select the appropriate result from the recommendations. A single misclassification in this work field can cost an exporter or importer thousands, even millions of dollars and is not a risk worth taking. Our system ensures that classification error during auto-classification does not create that issue, which is why the involvement of a human expert is of significant importance. Every selection the human expert makes results in a clean and labeled dataset over time.

We ensured scalability by introducing GPU accelerated support using TensorFlow-GPU and NVIDIA CUDA for fast processing of EDI trade transactions. All of the models we used took less than a minute to produce results for ten thousand transactions. In terms of speed STSB-RoBERTa using NVIDIA CUDA was the fastest, however, Google's USE using TensorFlow-GPU was only 2 seconds behind in processing ten thousand transactions. We have tested them on 2 physical environments and one cloud environment. Furthermore, we also accounted for interpretability since the early stages of our approach. The evaluation criteria we defined ensures that the client understands how the system works and how to properly interpret results.

The challenge of HS code classification is a global one. Our system ensures a safer classification approach to address that challenge. It can also be utilized in other application areas with same parameters, mainly for unstructured short English texts with large number of classes. One of those application areas can be the medical domain, specifically clinical coding [85].

Finally, due to the technical feasibility our approach offers, it was adopted by the client in their production system.

7.1 Limitations

Due to the nature of the approach being human-assisted, it relies on the knowledge of the human expert. The approach minimizes manual human effort to minimize error but is not immune to one. Since we had very limited data, the approach may not work as well for other shippers. That would require more data from the specific shippers. To keep document processing times even, the priority of better hardware increases with larger volumes of data to process bulk amounts of trade documents at once. The approach is built specifically for very noisy and unstructured data sets.

7.2 Future Work

As a future scope of the current work, it might be possible to build a better scoring system to confidently classify HS commodity codes.

Further efforts can be spent on performance improvement and auto-assignment of commodity codes without recommendations by using a large amount of historical shipping data and developing a confidence scoring system. Adopting the current recommendation-based approach in the working environment and working with more data it is possible for the client to develop a clean labeled data set. Each choice the human expert makes from the commodity recommendations can be fed to develop a labeled data set for future to build our own classifier with a confidence scoring system; i.e., we can associate a confidence score for each classification and possibly the lower scores will need the human expert's attention, thus reducing the effort considerably and also have a way of improving the score with each selection. However, it will require a large amount of data from the individual shippers considering the total number of classes being more than six thousand and also the variance of information in shipping manifests.

Since our data amount was very limited, we did not have enough repeated commodities to consider seasonality of the traded products which can be an important aspect in the classification or detection of misclassification. For example, mangoes are traded extensively during the Summer season.

Also, with more data, it is possible to create shipper profiles that can help boost the confidence scoring for classification and towards reducing misclassification of HS commodity codes. For example, a company whose primary product is water-pump probably will not ship oranges.

In addition to the above, translation or support for languages other than English in the trade documents such as French, Spanish, Chinese, Hindi, etc may also improve the end results considerably.

Bibliography

[1] Semantics3. Anatomy of HS codes, 2021. https://www.semantics3.com/blog/anatomy-of-hs-codes-6863f900adc1/, accessed on 2021-Jan-22.

[2] Jobisez LLC. Sample EDI, 2021. https://www.jobisez.com/edi/documents/example-edi.aspx?set-id=311, accessed on 2021-Jan-30.

[3] Dhyanesh Narayanan. Ashok Chandra. Machine learning and text analytics, 2020. https://docs.microsoft.com/en-us/archive/blogs/machinelearning/machine-learning-and-text-analytics, accessed on 21-Nov-2020.

[4] Yinfei Yang and Chris Tar. Advances in semantic textual similarity, 2021. https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html, accessed on 2021-Jan-23.

[5] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. pages 670–680, Sept 2017.

[6] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[7] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. pages 3973–3983, Jan 2019.

[8] Joydwip Mohajon towards data science. Confusion matrix for your multi-class machine learning model – a beginner's guide on how to calculate precision, recall, f1-score for a multi-class classification problem., 2020. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826, accessed on 2020-June-20.

[9] world bank.org. Canada trade summary 2019. https://wits.worldbank.org/CountryProfile/en/Country/CAN/Year/LTST/Summarytext, accessed on 2021-Feb-20.

[10] Trade.gov. Harmonized system (hs) codes, 2021. https://www.trade.gov/harmonized-system-hs-codes, accessed on 2021-June-26.

[11] Statistics Canada STATCAN. Commodity classification, 2021. https://www5.statcan.gc.ca/cimt-cicm/page-page?lang=eng&mode=commodityClassification, accessed on 2021-June-27.

[12] IBM. What is EDI?, 2021. https://www.ibm.com/topics/edi-electronic-data-interchange, accessed on 2021-June-26.

[13] WIKIPEDIA. Electronic data interchange, 2021. https://en.wikipedia.org/wiki/Electronic_data_interchange, accessed on 2021-June-26.

[14] Holm Kappler. Revering the trend: low cost and low risk methods for assuring paper duty payments. *World Customs Journal*, pages Vol.5, No.2, pp. 109 – 122, 2011.

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

[16] Wikipedia contributors. Combined nomenclature - Wikipedia, the free encyclopedia, 2019. https://en.wikipedia.org/wiki/Combined_Nomenclature, accessed on 21-Jan-2021.

[17] Wikipedia contributors. Taric code - Wikipedia, the free encyclopedia, 2019. https://en.wikipedia.org/wiki/TARIC_code, accessed on 21-Jan-2021.

[18] Irmak Aktan MTS Logistics. More than shipping - the importance of hs codes when importing or exporting goods, 2021. https://www.morethanshipping.com/the-importance-of-hs-codes-when-importing-or-exporting-goods/, accessed on 2021-Feb-26.

[19] Batarlienė Nijole and Aldona Jarašūnienė. Analysis of the accidents and incidents occurring during the transportation of dangerous goods by railway transport. *Transport*, 29, Oct 2014.

[20] Ron Triepels, Hennie Daniels, and Ad Feelders. Data-driven fraud detection in international shipping. *Expert Systems with Applications*, 99, June 2018.

[21] Theo Notteboom, Thanos Pallis, and Jean-Paul Rodrigue. Disruptions and resilience in global container shipping and ports: the covid-19 pandemic versus the 2008–2009 financial crisis. *Maritime Economics & Logistics*, Jan 2021.

[22] Worlds Customs Organization. Worlds customs organization - official web page, 2021. http://www.wcoomd.org/, accessed on 2021-Feb-20.

[23] Rodolfo Vazquez. HS code classification freeway: Take your exit, 2020. https://www.linkedin.com/pulse/hs-code-classification-freeway-take-your-exit-rodolfo-vazquez, accessed on 2020-Nov-23.

[24] Worlds Customs Organization. What is the harmonized system (hs)?, 2021. http://www.wcoomd.org/en/topics/nomenclature/overview/what-is-the-harmonized-system.aspx, accessed on 2021-March-15.

[25] Shaohua Du, Zhihao Wu, Huaiyu Wan, and YouFang Lin. *HScodeNet: Combining Hierarchical Sequential and Global Spatial Information of Text for Commodity HS Code Classification*, pages 676–689. May 2021.

[26] Kwang-Baek Kim, Minhwan Kim, and Young Woo. Recognition of shipping container identifiers using art2-based quantization and a refined rbf network. volume 4432, pages 572–581, April 2007.

[27] S.J. Barro-Torres, Tiago Fernández-Caramés, Miguel González-López, and Carlos Escudero. Maritime freight container management system using rfid. Sept 2010.

[28] Sergei Maruev, Dmitry Stefanovskiy, Aleksey Frolov, Alexander Troussov, and John Curry. Deep mining of custom declarations for commercial goods. *Procedia Economics and Finance*, 12:397–402, Dec 2014.

[29] Carl Steyn and Alta de Waal. Semi-supervised machine learning for textual anomaly detection. *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pages 1–5, 2016.

[30] Kaiqi Zhao, Lisi Chen, and G. Cong. Topic exploration in spatio-temporal document collections. In *SIGMOD '16*, 2016.

[31] Michael Johannesmeyer, Ashish Singhal, and Dale Seborg. Pattern matching in historical data. *AIChE Journal*, 48:2022 – 2038, Sept 2002.

[32] Philip Bernstein, Sergey Melnik, Michalis Petropoulos, and Christoph Quix. Industrial-strength schema matching. *SIGMOD Record*, 33:38–43, Dec 2004.

[33] Paolo Nesi, Gianni Pantaleo, and Marco Tenti. Geographical localization of web domains and organization addresses recognition by employing natural language processing, pattern matching and clustering. *Engineering Applications of Artificial Intelligence*, 51:202–211, May 2016.

[34] Kyungah Yang, Won-Jung Kim, Jae Yang, and Young Kim. Ontology matching for recommendation of hs code. Dec 2013.

[35] Guo Li and Na Li. Customs classification for cross-border e-commerce based on text-image adaptive convolutional neural network. *Electronic Commerce Research*, 19, Dec 2019.

[36] Stephen Peters and Howard Shrobe. Using semantic networks for knowledge representation in an intelligent environment. Feb 2003.

[37] Alok Pal and Diganta Saha. Word sense disambiguation: A survey. *International Journal of Control Theory and Computer Modeling*, 5, Aug 2015.

[38] Junaed Hossain, Nor Fazlida Mohd Sani, Lilly Affendey, Iskandar Ishak, and Khairul Kasmiran. Semantic schema matching approaches: A review. *Journal of Theoretical and Applied Information Technology*, 62:139–147, April 2014.

[39] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[40] Mario Jarmasz and Stan Szpakowicz. Roget's thesaurus: a lexical resource to treasure. April 2012.

[41] Rebecca Green, Bonnie Dorr, and Ps Resnik. Inducing frame semantic verb classes from wordnet and ldoce. pages 375–382, Jan 2004.

[42] Xavier Schmitt, Sylvain Kubler, Jérémy Robert, Mike Papadakis, and Yves LeTraon. A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. pages 338–343, Oct 2019.

[43] Nilanjana Nath, Shreekant Jha, Janki M, and Syedibrahim S.p. A survey paper on elastic search similarity algorithm. *Asian Journal of Pharmaceutical and Clinical Research*, 10:361, July 2017.

[44] Sebastian Raschka. Naïive Bayes and text classification I — introduction and theory. Oct 2014.

[45] Yuqian Jiang, Huaizhong Lin, Xuesong Wang, and Dongming Lu. A technique for improving the performance of naive bayes text classification. pages 196–203, Jan 2011.

[46] Md. Zahidul Islam, Jixue Liu, Jiuyong Li, Lin Liu, and Wei Kang. A semantics aware random forest for text classification. pages 1061–1070, Nov 2019.

[47] Ameni Bouaziz, Christel Dartigues-Pallez, Celia da Costa Pereira, Frederic Precioso, and Patrick Lloret. Short text classification using semantic random forest. pages 288–299, Sept 2014.

[48] Atreya Basu, Carolyn Watters, and Michael Author. Support vector machines for text categorization. page 103, Jan 2003.

[49] Rudolf Mathar, Gholamreza Alirezaei, Emilio Balda, and Arash Behboodi. *Support Vector Machines*, pages 83–105. Sept 2020.

[50] Qian Li, Hao Peng, Jianxin Li, Congyin Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. A survey on text classification: From shallow to deep learning. *ArXiv*, abs/2008.00364, 2020.

[51] Ye Zhang, Md Mustafizur Rahman, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, Aaron Angert, Edward Banner, Vivek Khetan, Tyler McDonnell, An Nguyen, Dan Xu, Byron Wallace, and Matthew Lease. Neural information retrieval: A literature review. Nov 2016.

[52] Peng Jin, Yue Zhang, Xingyuan Chen, and Yunqing Xia. Bag-of-embeddings for text classification. In *IJCAI*, 2016.

[53] Davide Liga and M. Palmirani. Transfer learning with sentence embeddings for argumentative evidence classication. In *CMNA@COMMA*, 2020.

[54] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. March 2018.

[55] Marine Traffic. Maersk palermo summary - marinetraffic, 2021. `https://www.marinetraffic.com/en/ais/details/ships/shipid:224755/mmsi:636020264/imo:9168207/vessel:MAERSK_PALERMO`, accessed on 16-Feb-2021.

[56] Maersk Line. Maersk - logistics made easy through digital solutions, 2021. `https://www.maersk.com/`, accessed on 16-Feb-2021.

[57] George Petras, Stephen J. Beard, Ramon Padilla and Shawn J. Sullivan, USA TODAY. ..evergreen's ship get stuck in the suez canal.., 2021. `https://www.usatoday.com/in-depth/graphics/2021/03/26/how-evergreens-ship-got-stuck-in-the-suez-canal/7010375002/`, accessed on 31-March-2021.

[58] Jing li, Aixin Sun, Ray Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1, March 2020.

[59] The Apache Software Foundation. Apache opennlp, 2020. `https://opennlp.apache.org/`, accessed on 20-Nov-2020.

[60] Stanford NLP Group. Stanford named entity recognizer (ner), 2020. `https://nlp.stanford.edu/software/CRF-NER.html`, accessed on 20-Nov-2020.

[61] Explosion.ai. Spacy entity recognizer, 2020. `https://spacy.io/api/entityrecognizer`, accessed on 20-Nov-2020.

[62] A. McCallum and K. Nigam. A comparison of event models for Naïve Bayes text classification. In *AAAI 1998*, 1998.

[63] Lior Rokach and Oded Maimon. *Decision Trees*, volume 6, pages 165–192. Jan 2005.

[64] Corinna Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 2004.

[65] Hajah Sueno, Bobby Gerardo, and Ruji Medina. Multi-class document classification using support vector machine (SVM) based on improved Naïve Bayes vectorization technique. *International Journal of Advanced Trends in Computer Science and Engineering*, 9:3937, 2020.

[66] Yang Li and Tao Yang. *Word Embedding for Understanding Natural Language: A Survey*, volume 26. May 2017.

[67] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013, Jan 2013.

[68] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[69] Ryan Kiros, Yukun Zhu, R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015.

[70] Aikaterini-Lida Kalouli, Annebeth Buis, Livy Real, Martha Palmer, and Valeria de Paiva. Explaining simple natural language inference. In *LAWACL*, 2019.

[71] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[72] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[73] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. June 2015.

[74] Rizvi Analytics Vidhya. Demystifying BERT: A comprehensive guide to the groundbreaking NLP framework, 2020. `https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework`, accessed on 2-Dec-2020.

[75] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[76] Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[77] Common Crawl. Common crawl news dataset, 2021. `https://commoncrawl.org/2016/10/news-dataset-available/`, accessed on 2021-6-12.

[78] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*, 2018.

[79] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, 2020. Association for Computational Linguistics.

[80] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. In *NeurIPS*, 2019.

[81] Gregory R. Koch. Siamese neural networks for one-shot image recognition. 2015.

[82] Caliber Inc. EDI parser, 2020. `http://ediparser.caliber.com`, accessed on 2020-07-30.

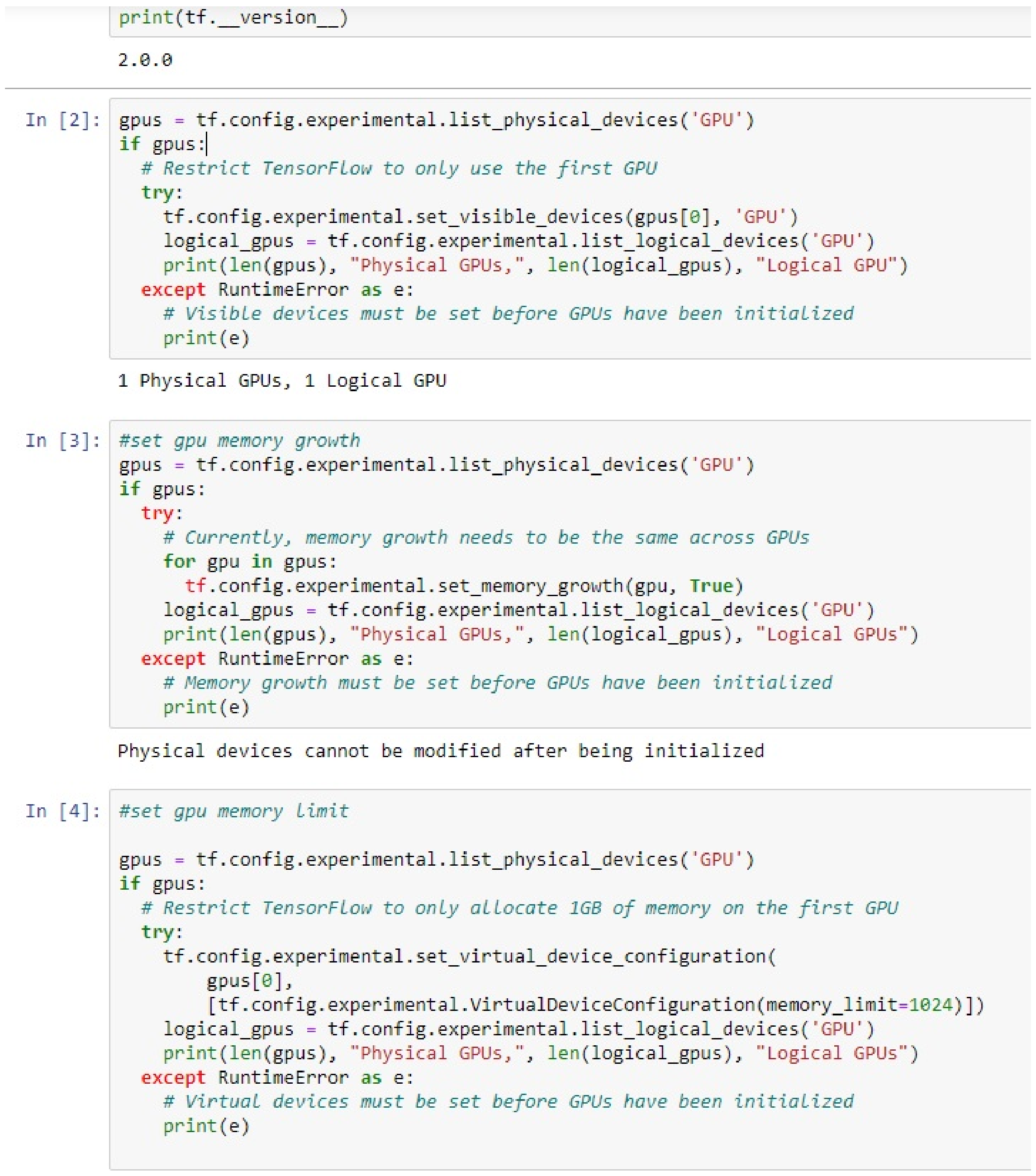
[83] DataLadder.com. Data ladder, 2020. `DataLadder.com`, accessed on 2020-Sept-20.

[84] OPENPRISE INC. Openprise, 2020. `https://www.openprisetech.com/`, accessed on 2020-Sept-29.

[85] Wikipedia contributors. Clinical coder, 2021. `https://en.wikipedia.org/wiki/Clinical_coder`, accessed on 2021-6-17.

Appendix A

Initializing TensorFlow-GPU on a Local PC – Code Snippet

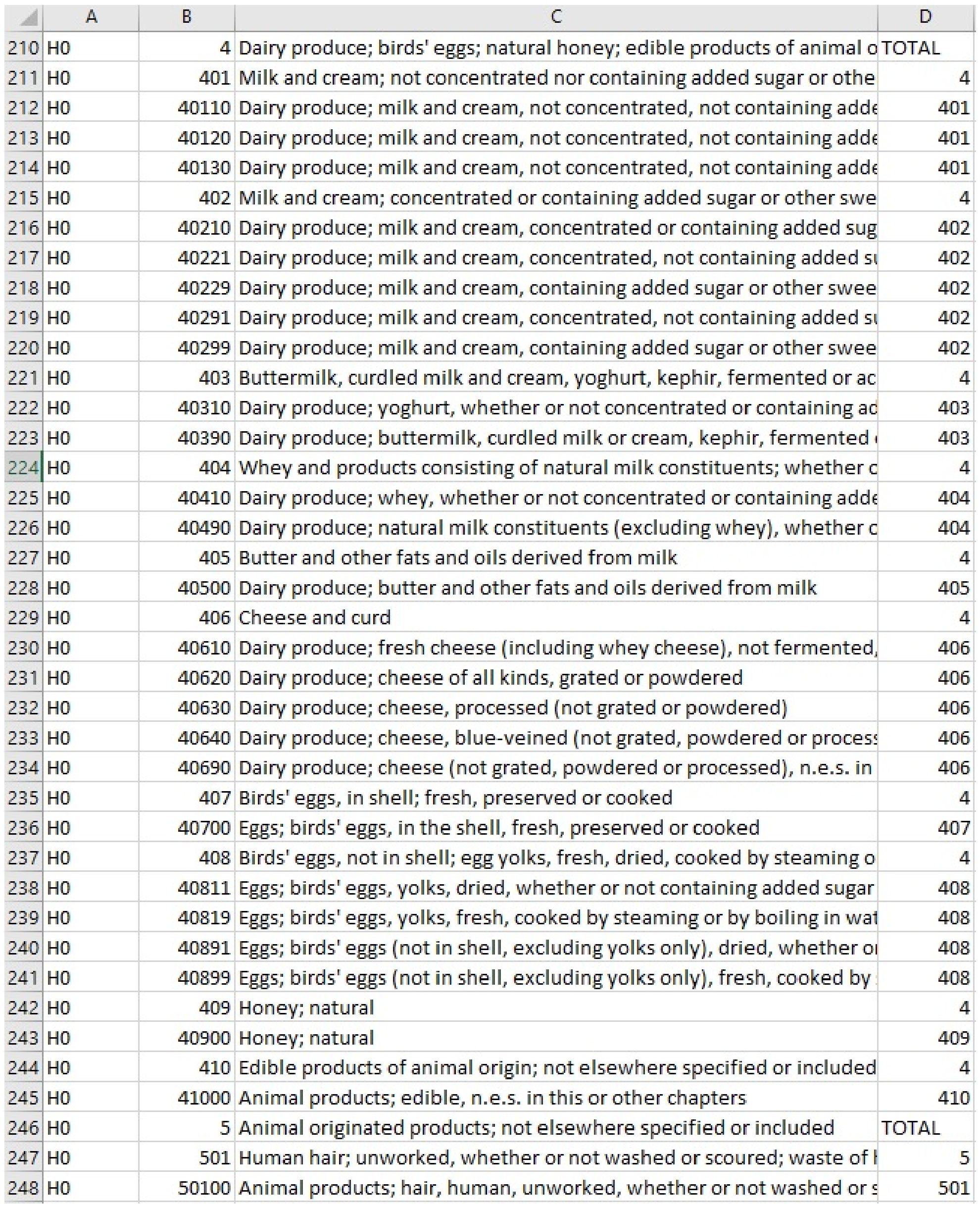


```
         print(tf.__version__)

         2.0.0

In [2]: gpus = tf.config.experimental.list_physical_devices('GPU')
        if gpus:
          # Restrict TensorFlow to only use the first GPU
          try:
            tf.config.experimental.set_visible_devices(gpus[0], 'GPU')
            logical_gpus = tf.config.experimental.list_logical_devices('GPU')
            print(len(gpus), "Physical GPUs,", len(logical_gpus), "Logical GPU")
          except RuntimeError as e:
            # Visible devices must be set before GPUs have been initialized
            print(e)

        1 Physical GPUs, 1 Logical GPU

In [3]: #set gpu memory growth
        gpus = tf.config.experimental.list_physical_devices('GPU')
        if gpus:
          try:
            # Currently, memory growth needs to be the same across GPUs
            for gpu in gpus:
              tf.config.experimental.set_memory_growth(gpu, True)
            logical_gpus = tf.config.experimental.list_logical_devices('GPU')
            print(len(gpus), "Physical GPUs,", len(logical_gpus), "Logical GPUs")
          except RuntimeError as e:
            # Memory growth must be set before GPUs have been initialized
            print(e)

        Physical devices cannot be modified after being initialized

In [4]: #set gpu memory limit

        gpus = tf.config.experimental.list_physical_devices('GPU')
        if gpus:
          # Restrict TensorFlow to only allocate 1GB of memory on the first GPU
          try:
            tf.config.experimental.set_virtual_device_configuration(
                gpus[0],
                [tf.config.experimental.VirtualDeviceConfiguration(memory_limit=1024)])
            logical_gpus = tf.config.experimental.list_logical_devices('GPU')
            print(len(gpus), "Physical GPUs,", len(logical_gpus), "Logical GPUs")
          except RuntimeError as e:
            # Virtual devices must be set before GPUs have been initialized
            print(e)
```

Figure A.1: Initialize, set memory growth and limit of TF-GPU

Appendix B

Example Screenshot of a Section From the UN Comtrade Sheet

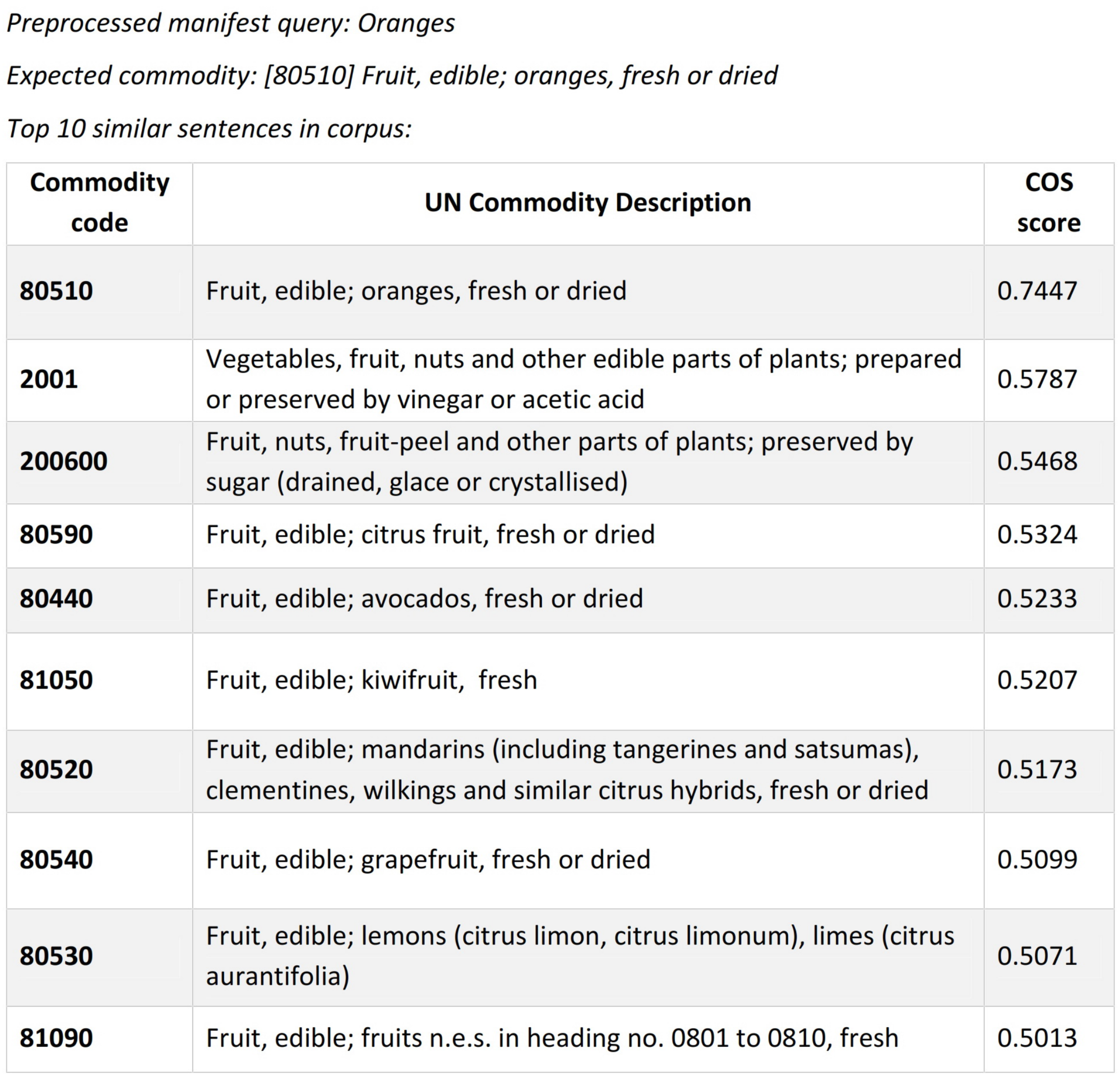


	A	B	C	D
210	H0	4	Dairy produce; birds' eggs; natural honey; edible products of animal o	TOTAL
211	H0	401	Milk and cream; not concentrated nor containing added sugar or othe	4
212	H0	40110	Dairy produce; milk and cream, not concentrated, not containing adde	401
213	H0	40120	Dairy produce; milk and cream, not concentrated, not containing adde	401
214	H0	40130	Dairy produce; milk and cream, not concentrated, not containing adde	401
215	H0	402	Milk and cream; concentrated or containing added sugar or other swe	4
216	H0	40210	Dairy produce; milk and cream, concentrated or containing added sug	402
217	H0	40221	Dairy produce; milk and cream, concentrated, not containing added su	402
218	H0	40229	Dairy produce; milk and cream, containing added sugar or other swee	402
219	H0	40291	Dairy produce; milk and cream, concentrated, not containing added su	402
220	H0	40299	Dairy produce; milk and cream, containing added sugar or other swee	402
221	H0	403	Buttermilk, curdled milk and cream, yoghurt, kephir, fermented or ac	4
222	H0	40310	Dairy produce; yoghurt, whether or not concentrated or containing ac	403
223	H0	40390	Dairy produce; buttermilk, curdled milk or cream, kephir, fermented	403
224	H0	404	Whey and products consisting of natural milk constituents; whether c	4
225	H0	40410	Dairy produce; whey, whether or not concentrated or containing adde	404
226	H0	40490	Dairy produce; natural milk constituents (excluding whey), whether c	404
227	H0	405	Butter and other fats and oils derived from milk	4
228	H0	40500	Dairy produce; butter and other fats and oils derived from milk	405
229	H0	406	Cheese and curd	4
230	H0	40610	Dairy produce; fresh cheese (including whey cheese), not fermented,	406
231	H0	40620	Dairy produce; cheese of all kinds, grated or powdered	406
232	H0	40630	Dairy produce; cheese, processed (not grated or powdered)	406
233	H0	40640	Dairy produce; cheese, blue-veined (not grated, powdered or process	406
234	H0	40690	Dairy produce; cheese (not grated, powdered or processed), n.e.s. in	406
235	H0	407	Birds' eggs, in shell; fresh, preserved or cooked	4
236	H0	40700	Eggs; birds' eggs, in the shell, fresh, preserved or cooked	407
237	H0	408	Birds' eggs, not in shell; egg yolks, fresh, dried, cooked by steaming o	4
238	H0	40811	Eggs; birds' eggs, yolks, dried, whether or not containing added sugar	408
239	H0	40819	Eggs; birds' eggs, yolks, fresh, cooked by steaming or by boiling in wat	408
240	H0	40891	Eggs; birds' eggs (not in shell, excluding yolks only), dried, whether o	408
241	H0	40899	Eggs; birds' eggs (not in shell, excluding yolks only), fresh, cooked by	408
242	H0	409	Honey; natural	4
243	H0	40900	Honey; natural	409
244	H0	410	Edible products of animal origin; not elsewhere specified or included	4
245	H0	41000	Animal products; edible, n.e.s. in this or other chapters	410
246	H0	5	Animal originated products; not elsewhere specified or included	TOTAL
247	H0	501	Human hair; unworked, whether or not washed or scoured; waste of h	5
248	H0	50100	Animal products; hair, human, unworked, whether or not washed or s	501

Figure B.1: Screenshot of a section from the official UN commodity classification sheet

Appendix C

Additional Example Result for Commodity ‘Oranges’



Preprocessed manifest query: Oranges

Expected commodity: [80510] Fruit, edible; oranges, fresh or dried

Top 10 similar sentences in corpus:

Commodity code	UN Commodity Description	COS score
80510	Fruit, edible; oranges, fresh or dried	0.7447
2001	Vegetables, fruit, nuts and other edible parts of plants; prepared or preserved by vinegar or acetic acid	0.5787
200600	Fruit, nuts, fruit-peel and other parts of plants; preserved by sugar (drained, glace or crystallised)	0.5468
80590	Fruit, edible; citrus fruit, fresh or dried	0.5324
80440	Fruit, edible; avocados, fresh or dried	0.5233
81050	Fruit, edible; kiwifruit, fresh	0.5207
80520	Fruit, edible; mandarins (including tangerines and satsumas), clementines, wilkings and similar citrus hybrids, fresh or dried	0.5173
80540	Fruit, edible; grapefruit, fresh or dried	0.5099
80530	Fruit, edible; lemons (citrus limon, citrus limonum), limes (citrus aurantifolia)	0.5071
81090	Fruit, edible; fruits n.e.s. in heading no. 0801 to 0810, fresh	0.5013

Figure C.1: Example result for a pre-processed commodity description ‘Oranges’