Baltic J. Modern Computing, Vol. 8 (2020), No. 4, pp. 698–718
https://doi.org/10.22364/bjmc.2020.8.4.13

# Application of Machine Learning for Assessment of HS Code Correctness

Margarita SPICHAKOVA, Hele-Mai HAAV

Department of Software Science, Tallinn University of Technology.
Akadeemia tee 15a, 12618 Tallinn, Estonia

margarita.spitsakova@taltech.ee, helemai@cs.ioc.ee

**Abstract.** Manual assessment of the correctness of Harmonized System codes of goods is very error-prone and time demanding task taking into account the dramatically growing amounts of cross-border trade. The paper provides an automated solution to this problem by applying machine learning methods to assess the correctness of Harmonized System codes. We use machine learning for providing predictions and recommendations of Harmonized System codes on the basis of a model learned from the textual descriptions of the products. In order to assess the correctness Harmonized System codes of goods we introduce a novel combined similarity measure based on cosine similarity of texts and semantic similarity of Harmonized System codes calculated according to their taxonomy. We also present and prove the properties of this new similarity measure. We test our method on the real open source data set of Bill of Lading Summary 2017.

**Keywords:** machine learning, doc2vec, harmonized system taxonomy, cosine similarity of text, semantic similarity.

## 1 Introduction

The Harmonized System (HS) code nomenclature created by the World Customs Organization (WCO) already in 1988 (WEB, c) is a 6-digit standardized international numerical code to represent and identify goods for worldwide trade. It is used to derive the tariff to be assessed. Misclassification of goods according to the HS is common and incorrect HS codes create a huge additional cost for retailers and e-marketplaces. In addition, the misclassification of goods leads to duty underpayments. In 2017, the European Court of Auditors has reported that widely applied forms of evasion of customs duty payments are undervaluation, misclassification by shifting to a product classification with a lower duty rate and the wrong description of the origin of imported goods (WEB, 2017a). For example, the potential losses of customs duties were calculated to be close to 2 billion euro for the period 2013–2016 due to undervaluation of imports of textiles and footwear from China into the UK (WEB, 2017a).

In order to cope with growing fraud, the European Commission has announced a legal Value Added Tax (VAT) reform (to be forced by 2021) that aims to react to those problems, increase tax compliance and make certain procedures easier for companies selling via cross-border e-commerce (WEB, 2017b). Although the directive will be in force in 2021, tax and customs authorities and other relevant stakeholders systems are not ready yet and need automated solutions for product classification as well as assessment of the correctness of HS codes for fraud detection.

Customs agents use the product's origin and value together with the HS code to derive the tariff to be assessed. However, the correct classification of HS codes remains a challenging task (Li and Li, 2019). Misclassification of products can have several reasons as listed in (Kappler, 2011). First, the HS nomenclature and the rules governing the classification process are very complex. Second, there is a terminological and the semantic gap between product descriptions in the HS nomenclature and goods descriptions in trade (i.e. commercial terms). This leads to the problem that simple text search that is currently used by many HS code databases and lookup systems cannot help traders to locate the relevant HS codes because of the difference and semantic disambiguation between the structured descriptions of the HS nomenclature and the text descriptions used during the trade process.

For solving the problem of misclassification of products many researchers propose automated systems for the classification of HS codes, considering this as a multi-class classification problem and using several machine learning approaches (Ding et al., 2015), (Li and Li, 2019), (Turhan et al., 2015) or provide expert systems like 3CE HS code verification system (WEB, g) or automated neural network based tools for product categorization and mapping to HS codes (WEB, f).

In contrast, we aim in this paper at an automated assessment of the correctness of HS codes that are already classified and received from e-marketplaces or retailers to customs or logistic companies. In addition to assessment, we also provide some recommendations for corrections of HS codes in the cases of low assessment scores.

The main contribution of the paper is a new machine learning based method for automated assessment of the correctness of HS codes. It is built upon a new proposed combined similarity measure of concepts. Maximality, positiveness and symmetry properties are proved for this new similarity measure making it possible to be used in many other similar applications. In this paper, introduced combined similarity measure shows how good the textual description of the given HS code matches with the HS code's position in the HS code taxonomy (classification). The major advantage of our approach compared to pure text based learning approaches is that we may derive additional knowledge from HS code structure (taxonomy) in order to complement short textual descriptions of goods that alone provide insufficient knowledge for automated HS code correctness assessment.

This paper is an extended version of our previous paper (Spichakova and Haav, 2020). Main extensions include proofs of properties of our new combined similarity measure and evaluation of assignments of the HS code correctness assessment scores.

The method is evaluated on the real open source data set of Bill of Lading Summary 2017 (WEB, a) that after filtering contains approximately 1.2 million rows with product

descriptions and their corresponding HS codes. Gensim Python library (WEB, b) is used for the implementation of the method.

The paper is structured as follows. In Section 2, a review of related approaches is presented. Section 3 gives some preliminary definitions and explanations to be used in the rest of the paper. In Section 4, we introduce a new combined similarity measure that forms the basis of the HS code correctness assessment method provided in this paper. Section 5 is devoted to an overview of our method and its components. In Section 6, we provide an evaluation of the method and descriptions of experiments. Section 7 concludes the paper.

## 2 Related works

Most of the related work is devoted to automated commodity classification from the seller's perspective by using different machine learning approaches (Ding et al., 2015), (Li and Li, 2019), (Turhan et al., 2015). Our work has a perspective on fraud detection as a part of risk management systems.

Fusing textual descriptions of goods and visual data about products in order to automatically classify and recommend HS codes is a rather new approach found in the literature (Li and Li, 2019), (Turhan et al., 2015). Li and Li propose text–image dual convolutional neural network model for automated customs classification. The experiments of this work have been conducted on a limited manually tagged data set of 10000 records including data only for four classes showing that fusion model works better than non-fusion model.

Another work in (Turhan et al., 2015) combines topic maps, visual feature extraction and matching techniques. The method is evaluated on 4494 records. However, their result for combined recommendations of HS codes has an accuracy rate of about 78 percent.

In contrast to the works above, we do not use visual information but the structure of HS code taxonomy (ontology) as complementary knowledge to pure text.

In (Yang et al., 2013), a HS code recommendation approach to a product is presented. Their system is based on three types of ontologies: the HS code ontology, the product classification ontology and domain ontology. This is only related work where the HS code ontology is used. However, their HS code ontology is different from our HS code ontology in two aspects: we do not use any textual descriptions of ontology concepts as they are already available in the HS code nomenclature and in addition, we capture also sections in our HS code ontology.

In (Wang et al., 2015) an interesting method for improving performance of short text classification is presented. Their method uses semantic clustering and Convolutional Neural Network (CNN) for short text classification. They first cluster word embeddings and discover semantic cliques. After that, semantic composition is performed over n-gram embeddings to detect candidate Semantic Units (SU) appearing in short texts. A set of candidate SUs that meet the given threshold are chosen to constitute semantic matrices, which are used as input for the CNN. The classification accuracy of their method compared to other methods is promising.

The work in (Ruder, 2020) is devoted to the problem of assignment of HS codes to products that can be seen as a multi-class text classification problem, where textual descriptions of products (used in cargo) become a basis for classification of products to different classes defined by HS codes. The main goal of this work is to evaluate the applicability and the efficiency of existing multi-class text classification methods on the HS code classification problem. This work uses the same real world data set as we have used for our experiments but from the period 2018-2020. The input feature vectors for classification algorithms were obtained by using term frequency-inverse document frequency (TF-IDF) (Salton and Buckley, 1988), Doc2Vec (Le and Mikolov, 2014), Word2Vec (Mikolov et al., 2013a), and GloVe (Pennington et al., 2014) methods. The work presents results of a set of comprehensive experiments of application of a number of machine learning algorithms. The received feature vectors were used as input to Rocchio classification algorithm (Rocchio, 1965), Multinomial Logistic Regression (MLR) (Hosmer et.al., 2013), Multinomial Nave Bayes (MNB) (Larson, 2010), K–nearest Neighbours (KNN) (Guo et al., 2003), Decision Tree (Noormanshah et al., 2018), Random Forests (Xu, et al, 2012), Support Vector Machine (SVM) (Manevitz et al., 2001), Deep Neural Network (DNN) (Lecun, et al., 2015), and CNN (Jaderberg, et al., 2016) classifiers. As a result, the DNN classifier with TF-IDF extracted features had the highest F-1 weighted average score of 61 percent and 62 percent accuracy. It is much better accuracy than averagely reached today by human work.

## 3 Preliminaries

### 3.1 HS code definition and classification taxonomy

The HS is used by 179 countries covering about 98 percent of world trade for the assessment of customs duties and the collection of statistical data (WEB, d). The HS nomenclature classifies products into 21 sections, 96 chapters, headings and subheadings giving possibility totally to classify about 8547 different classes (groups) of items (Li and Li, 2019).

For example, the correct HS code for the product described as ”laptop computer” is 847130. In the HS code nomenclature it has the explanation as follows: ”Portable automatic data processing machines, weighing not more than 10 kg, consisting of at least a central processing unit, a keyboard and a display” (WEB, e).

HS code digits are divided to three pairs, where the pairs of digits are referred to as Chapter, Heading and Subheading respectively. The HS code nomenclature can be represented as an ontology that includes only taxonomic (is-a) relationships and can be seen as a directed acyclic graph (DAG) (it is a tree as one HS code cannot belong to many headings at the same time) as shown in Fig. 1. In addition to the classes mentioned above, we add also classes that denote sections to our HS code taxonomy (ontology). This is because of the ontology will be used for the calculation of semantic similarity of HS codes (see Section 3.2.2). More hierarchy levels give more fine grained values for semantic similarity measures. The HS code classification is the process of finding the most specific description in the HS code taxonomy for the goods to be classified. For example, according to the taxonomy shown in Fig. 1, a product described as ”laptop

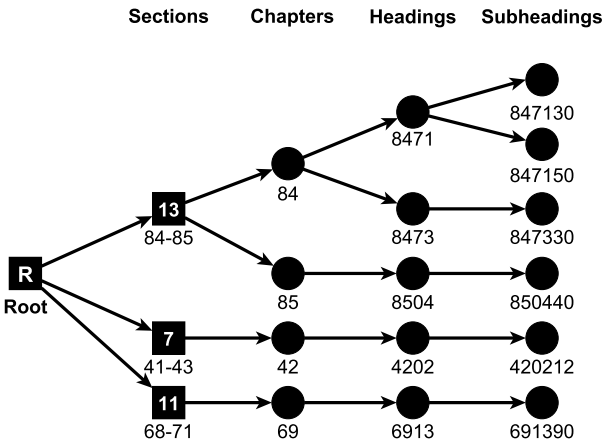


**Fig. 1.** An excerpt of HS code taxonomy.

computer" is classified to have the HS code 847130 (i.e. belongs to the subheading 847130).

### 3.2 Similarity measures

Similarity measures are widely used in many fields like Information Retrieval, text matching, ontology matching, machine learning etc. Generally, a similarity between a pair of entities is defined as follows (Euzenat and Shvaiko, 2013):

**Definition 1.** (Similarity) Given a set $\mathbb{O}$ of entities, a similarity $s : \mathbb{O} \times \mathbb{O} \rightarrow \mathbb{R}$ is a function from a pair of entities to a real number expressing the similarity between two objects such that

$$\forall x \in \mathbb{O},\ \forall y \in \mathbb{O},\ s(x, y) \geq 0 \text{ (positiveness)} \tag{1}$$

$$\forall x \in \mathbb{O},\ \forall y \in \mathbb{O},\ \forall z \in \mathbb{O},\ s(x, x) \geq s(y, z) \text{ (maximality)} \tag{2}$$

$$\forall x \in \mathbb{O},\ \forall y \in \mathbb{O},\ s(x, y) = s(y, x) \text{ (symmetry)} \tag{3}$$

The measures are often normalised, especially if the similarity of different kinds of entities must be compared. Reducing each value to the same scale in proportion to the size of the considered space is the common way to normalise.

**Definition 2.** (Normalised similarity) A similarity is said to be normalised if it ranges over the unit interval of real numbers $[0, 1]$.

In this work, we are interested in two types of similarity measures as follows: text based (or token-based) similarity and semantic (or ontology based) similarity.

**3.2.1 Text based similarity measures.** Descriptions of goods are short texts. Textual entities can be transformed into vectors to make it possible to apply usual metric space distances for similarity calculation. Basically, there are two approaches for computing feature vectors for textual entities as follows: using bag-of-words and TF-IDF methods or using word embeddings by applying Word2Vec (Mikolov et al., 2013a), Doc2Vec (Le and Mikolov, 2014) or GloVe (Pennington et al., 2014).

For example, Word2Vec (Mikolov et al., 2013a), (Mikolov et al., 2013b) is used to generate representation vectors from words. In this case, a word vector is a representation of text in a form of numerical vector. Word2Vec model embeds words in a lower-dimensional vector space using a neural network. The result is a set of words and vectors where vectors close together in vector space have similar meanings based on context, and vectors distant to each other have differing meanings. The textual similarity of words can be measured as cosine similarity that measures the cosine of the angle between two word vectors.

In our case, a document is a short description of some product (commodity). Therefore, we use the Doc2Vec model (Le and Mikolov, 2014) that extends the idea of Word2Vec (Mikolov et al., 2013a) to documents and enables to generate representation vectors for documents (e.g. paragraphs, sentences, etc) .

We measure the similarity between textual descriptions using cosine similarity between document vectors computed by Doc2Vec model.

**Definition 3.** (Cosine similarity of documents) Given $\vec{x}$ and $\vec{y}$, the representation vectors corresponding to documents $x$ and $y$ in vector spaces $\mathbb{V}$, the cosine similarity of documents $x$ and $y$ is the function $CosSim : \mathbb{V} \times \mathbb{V} \rightarrow [0, 1]$, such that

$$CosSim(x, y) = \frac{\sum_{i=1}^{|\mathbb{V}|} (\vec{x}_i \cdot \vec{y}_i)}{\sqrt{\sum_{i=1}^{|\mathbb{V}|} (\vec{x}_i)^2} \cdot \sqrt{\sum_{i=1}^{|\mathbb{V}|} (\vec{y}_i)^2}} \tag{4}$$

For example (see Fig. 2), cosine similarity between the vectors of the texts "laptop computer" and "laptop" is 0.886 but between "laptop computer" and "laptop bag" it is 0.864 (see also Table 2).

The cosine similarity measure fulfils all the properties of normalized similarity measure given in Definition 1.

The $CosSim$ takes values form the interval of $[0, 1]$ i.e. positiveness (in Equation 1) is fulfilled.

Maximality property is also fulfilled (in Equation 2) as

$$CosSim(x, x) = 1 \tag{5}$$

Finally, cosine is symmetric and therefore symmetry property (in Equation 3) holds true as follows:

$$CosSim(x, y) = CosSim(y, x) \tag{6}$$

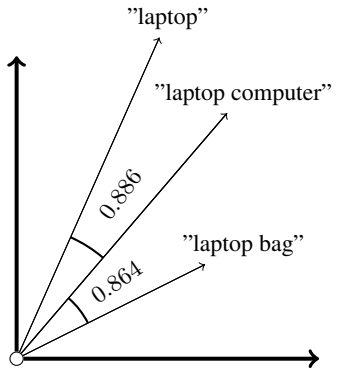


**Fig. 2.** Cosine similarity example

**3.2.2 Semantic similarity measures.** HS codes are arranged into taxonomy as we have seen above (see Fig. 1). This calls for applying a semantic distance method that is based on the structural knowledge in ontology (taxonomy). There are several measures developed for measuring semantic similarity of concepts in the ontology. A good overview is provided in (Sanchez et al., 2012), (Shenoy et al., 2012), (Euzenat and Shvaiko, 2013) on various approaches of calculating semantic similarity.

In this paper, we apply a method suggested in (Wu and Palmer, 1994) that takes into account the relative depth in the taxonomy of the concepts. Accordingly, equally distant pairs of concepts belonging to an upper level of taxonomy should be considered less similar than those belonging to a lower level. Originally, Wu and Palmer used their method in the domain of machine translation but nowadays, their method is widely used in many cases, where taxonomy depth needs to be taken account. Their similarity computation is based on the edge counting method.

**Definition 4.** (Wu and Palmer semantic similarity) The conceptual (semantic) similarity between ontology concepts $C_1$ and $C_2$ is defined as

$$WPSim = \frac{2 \cdot D}{D_1 + D_2} \tag{7}$$

The distances $D_1$ and $D_2$ separate nodes $C_1$ and $C_2$ from the root node $R$ and the distance $D$ separates the closest common ancestor (the least common subsumer) of $C_1$ and $C_2$ from the root node $R$.

For example, if $C_1$ is code 847130 and $C_2$ is code 850440, then

$$WPSim(847130, 850440) = \frac{2 \cdot 1}{4 + 4} = 0.25 \tag{8}$$

This example shows that section numbers on level one match but chapter numbers on level 2 mismatch. Higher is semantic similarity value closer are the given HS codes in the HS code taxonomy.

The Wu and Palmer semantic similarity measure fulfills all the properties of normalized similarity measure given in Definition 1 i.e. positiveness (in Equation 1), maximality (in Equation 2) and symmetry (in Equation 3).

$WPSim$ takes values from the $[0, 1]$ interval (positiveness). Maximality property is also fulfilled as

$$WPSim(C_1, C_1) = 1 \tag{9}$$

$WPSim$ is also symmetric as the following equation holds

$$WPSim(C_1, C_2) = WPSim(C_2, C_1) \tag{10}$$

In $WPSim$ similarity measure formula, the order of the concepts is irrelevant and it uses only commutative operations (i.e. summation ) over distances of concepts (i.e. nodes).

A disadvantage of this measure is that all the semantic links have the same weight. On the other hand, according to (Shenoy et al., 2012) the measure is easy to calculate and remains as expressive as the other semantic similarity measures. Therefore, we base our combined similarity measure on it. However, there are other options, for example, to take into count a number of children of a particular parent node. We may consider this in future work.

## 4 A proposed combined similarity measure

### 4.1 A new combined similarity measure

In the process of finding an automated solution for the HS code correctness assessment problem, we had a working hypothesis that the similar textual descriptions of products should be related to the similar HS codes. We applied the Doc2Vec model (Le and Mikolov, 2014) to get the corresponding representation vectors for predicted documents in order to calculate cosine similarities of these documents (e.g. product descriptions). However, application of the Doc2Vec model has shown that not always similar descriptions are related to the same or similar HS code meaning that our first working hypothesis does not always hold true. One reason is that texts of product descriptions are too short.

This led us to the idea that we need to take into account the taxonomy of HS codes in order to understand how good is the matching between the given HS code and the text of the description of the corresponding product. Therefore, we set up a new hypothesis that predicted HS codes and an original HS code should be close to each other in the HS code taxonomy. In order to test this hypothesis, we developed an original hybrid approach to similarity measure calculation for assessment of the correctness of HS codes.

We called this similarity measure as a combined similarity measure of concepts because it combines the knowledge derived from the textual descriptions and the corresponding taxonomy.

The combined similarity of concepts ($CombSim$) is defined as follows.

**Definition 5.** (Combined similarity of concepts) Given concepts $C_1$ , $C_2$ and the corresponding documents $x$ , $y$ describing these concepts, the combined similarity of ontology concepts $C_1$ and $C_2$ is defined as follows:

$$CombSim(\{C_1, x\}, \{C_2, y\}) = CosSim(x, y) \cdot WPSim(C_1, C_2) \quad (11)$$

$(C_1, x)$ is a pair of a concept $C_1$ and its textual description x and $(C_2, y)$ is a pair of a concept $C_2$ and its textual description y.

This is a general definition. When applied to the HS taxonomy and the corresponding product descriptions, cosine similarity ($CosSim$) of textual descriptions of goods is calculated using paragraph vectors obtained from Doc2Vec model according to cosine similarity calculation (see Definition 3). However, other methods can be used for getting representation vectors of documents. Wu and Palmer semantic similarity measure ($WPSim$) is calculated on the basis of HS codes assigned to the products and the HS code classification taxonomy (see Definition 4).

For example, combined similarity for $C_1 = 847130$ and $C_2 = 847150$ with the corresponding documents $x =$ "laptop computer" and $y =$ "desktop laptop" is as follows:

$$\begin{aligned} &CombSim(\{847130, \text{"laptop computer"}\}, \{847150, \text{"desktop laptop"}\}) = \\ &= WPSim(847130, 847150) \cdot CosSim(\text{"laptop computer"}, \text{"desktop laptop"}) = \\ &= 0.75 \cdot 0.886 = 0.665 \end{aligned} \quad (12)$$

High $CombSim$ values indicate that the given HS code matches better with the given product description than those that have lower $CombSim$ values. This also means that $CosSim$ values can be high but $CombSim$ values for the same product may be not as $WPSim$ values of HS codes can be low.

In general, the proposed combined similarity measure can be used for other similar tasks, for example, for ontology matching.

### 4.2 Properties of combined similarity measure

In the following, we show that the provided new similarity measure fulfils the properties defined for normalized similarity measure given by the Definition 1 i.e. positiveness (in Equation 1), maximality (in Equation 2) and symmetry (in Equation 3).

Positiveness property is fulfilled as $CombSim$ values range in the $[0, 1]$ interval because $CosSim$ ranges from 0 to 1 and $WPSim$ values range also from 0 to 1. Consequently, their product cannot be negative.

Maximality property is also fulfilled as

$$CombSim(\{C_1, x\}, \{C_1, x\}) = 1 \quad (13)$$

$CombSim$ value is 1 (maximum similarity) only if the concepts are compared to themselves and they are described by the same document. $CosSim$ value is 1 if the two vectors related to documents (texts) are perfectly similar. $WPSim$ value is 1 if the concepts in the taxonomy are the same.

$CombSim$ is also symmetric as the following equation holds true:

$$CombSim(\{C_1, x\}, \{C_2, y\}) = CombSim(\{C_2, y\}, \{C_1, x\}) \quad (14)$$

Multiplication is a commutative operator applied in combined similarity formula to $CosSim$ and $WPSim$ similarity measures that are both symmetric. Consequently, the combined similarity measure accomplishes the symmetry property.

## 5 HS code correctness assessment method

### 5.1 An overview of the method

Our HS Code correctness assessment method is based on the application of combined similarity measure defined above (see Definition 5). A general scheme of our method is provided in Fig. 3 and the explanation of its components in the following.

As input to our method, we require that for each HS code to be assessed the corresponding product description (in trade) is made available. These data come from external systems or sources.

In addition, the method requires the availability of the HS code nomenclature (i.e. classification taxonomy) and the Doc2Vec model that is to be trained on the given set of textual descriptions of products as presented in Section 6.2.

As depicted in Fig. 1, the Doc2Vec model is used for calculation of the cosine similarity values and for predicting HS code values. Predicted HS code values are used for calculating semantic similarity according to $WPSim$ formula and the given HS code taxonomy.

The combined similarity measure calculation is then performed according to $CombSim$ formula on the basis of cosine similarity and the corresponding semantic similarity values. Finally, the combined similarity values are used for the calculation of HS code correctness assessment scores.

### 5.2 Calculating textual similarities and prediction of HS code values

In order to obtain representation vectors (features) for documents we use feature extraction based on document embedding (Le and Mikolov, 2014) that is based on word embedding technique provided in (Mikolov et al., 2013a).

Word embedding is a method that allows predicting surrounding words (word vectors) of a given word. The most well-known word embedding algorithm is Word2Vec (Mikolov et al., 2013a). The Word2Vec model has two variants as follows:

1. Skip-grams (SG) takes in pairs of words generated by sliding a window across text, and trains a neural network with one hidden layer based on the task of giving a probability distribution of nearby words to the input word.
2. Continuous-bag-of-words (CBOW) is a neural network with one hidden layer. The training task now uses the average of multiple input context words, rather than a single word as in skip-gram, to predict the centre word. The projection weights that turn one-hot words into averageable vectors, of the same width as the hidden layer, are interpreted as the word embeddings. So if the hidden layer has 300 neurons, this network will give us 300-dimensional word embeddings.

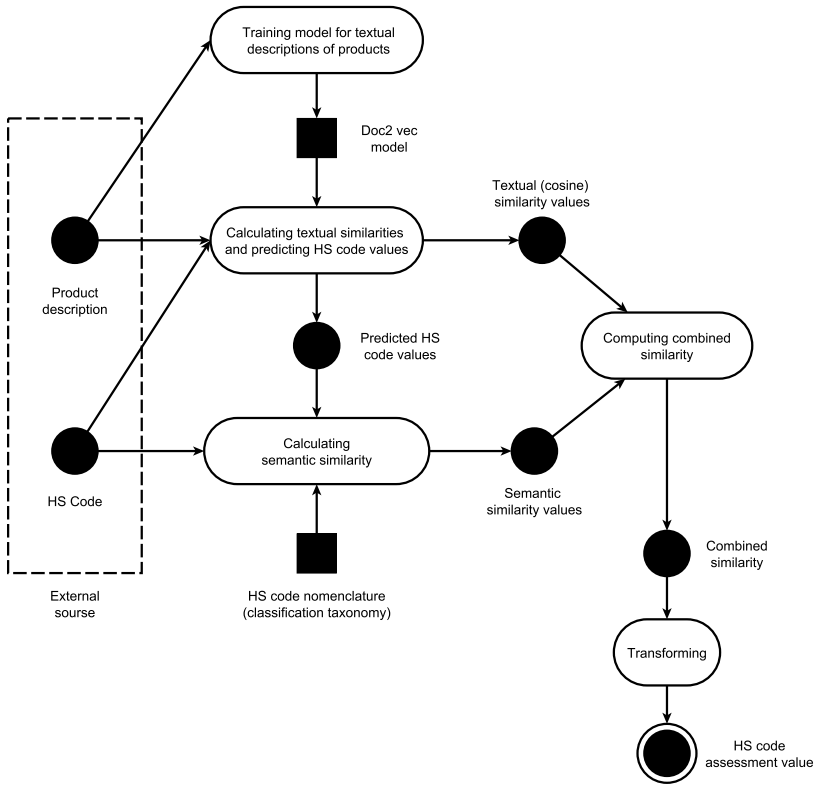


**Fig. 3.** Basic scheme of the method of HS code correctness assessment.

The Word2Vec model is effective in word-level representation but may come short in representing the semantics of a specific document. Therefore, we base our method on a document embedding algorithm Doc2Vec as presented in (Le and Mikolov, 2014). It is based on the word embedding technique, and it is used to extract surrounding word vectors that are specific to a document. In this model, the vector representation is trained to be useful for predicting words in a paragraph. The paragraph vector is concatenated with several word vectors from a paragraph in order to predict the following word in the given context (Le and Mikolov, 2014). The basic idea of this approach is that it acts as if a document has another floating word-like vector, which contributes to all training predictions and is updated like other word-vectors, but it is called as a doc-vector. There are two implementations of Doc2Vec as follows:

1. Paragraph Vector - Distributed Memory (PV-DM) is analogous to Word2Vec CBOW. The vectors are obtained by training a neural network on the task of predicting a centre word based an average of both context word-vectors and the full documents doc-vector.

2. Paragraph Vector - Distributed Bag of Words (PV-DBOW) is analogous to Word2Vec SG The doc-vectors are obtained by training a neural network on the synthetic task of predicting a target word just from the full documents doc-vector (Le and Mikolov, 2014).

In our method, we apply the Doc2Vec PV-DBOW variant implemented in Gensim Python library (WEB, b) for getting paragraph vectors for calculating cosine similarities of product descriptions and predicting HS codes. PV-DBOW is more suitable for short texts and therefore applicable in our case.

Specifically, in order to get representation vectors (features) for documents describing products we train the Doc2Vec model on the basis of the given data set (see Section 6.1).

Related to cosine similarity calculation of product descriptions we should make the following remarks concerning maximality property of cosine similarity measure.

Using Word2Vec, the cosine similarity calculation is related to word vectors and the similar words always are assigned the same representation vectors. Consequently, the maximum cosine similarity of similar words is always one.

However, the corresponding Doc2Vec model does not return for the similar input documents the same representation vectors (this is due to randomness). The model returns each time slightly different representation vectors. This influences to the value of the cosine similarity for the similar documents. First of all, it is not exactly one and as each time the model returns different representation vectors for the same document then maximum cosine similarity of the similar documents can vary but is still close to one. This feature of the Doc2Vec model does not influence its practical applicability in our case.

### 5.3 Assessment of HS code correctness on the basis of combined similarity

As a result of previous steps, we get for each predicted HS code its cosine similarity related to the given product description ($CosSim$), its semantic similarity related to the given HS code according to HS code taxonomy ($WPSim$) and its combined similarity ($CombSim$).

According to the HS code taxonomy in Fig. 1, we may pre-calculate $WPSim$ values for each of the mismatching taxonomy levels of the given HS code and predicted ones as the taxonomy has the tree structure with a fixed number of levels of 0 to 4. All possible $WPSim$ values for HS code taxonomy are provided in Table 1.

In order to produce HS code assessment scores we proceed as follows:

1. Sort a list of predicted HS codes in descending order of values of $CombSim$ and return $WPSim$ of the HS code which $CombSim$ is the best.
2. Convert returned $WPSim$ to the score as shown in Table 1. These scores can be easily interpreted also by humans.
3. Return the score as HS code assessment score ($HSScore$).

This scoring process assumes that non-existing HS codes (according to the HS code nomenclature) are detected before this procedure and removed from the input to this process.

**Table 1.** Interpretation of HS code assessment scores.

| $WPSim$ | $HSScore$ | Interpretation |
|---|---|---|
| 0 | 0 | Incorrect HS code; mismatching section |
| 0.25 | 1 | Incorrect HS code; matching section but mismatching chapter |
| 0.5 | 2 | Incorrect HS code; matching section and chapter but mismatching heading |
| 0.75 | 3 | Incorrect HS code; matching section, chapter and heading but mismatching subheading |
| 1 | 4 | Possibly correct HS code |

# 6 Method evaluation and experimental results

## 6.1 Data sources and data preprocessing

As a data source, we used Bill of Lading Summary 2017 (Bol2017) data set (WEB, a) published under CC BY-NC 4.0, which allows non-commercial use. The data set contains Bills of lading header information for incoming shipments regulated by the U.S. Customs and Border Protection's Automated Manifest System (WEB, a). The schema of the data set includes more than 20 different fields from what we are interested in the following two fields: Item Description and Harmonized Tariff Code. The latter is the code located in the Harmonized Tariff Schedule of the United States Annotated that describes the tariff number or Harmonized Tariff Schedule B that represents the commodity export (WEB, a). The data set contains the HTS code (Harmonized Tariff Schedule), which is used in the US. However, we are interested in the HS code, which is used by the WCO. HS code can be constructed as first 6 digits (out of 8–10 digits) of HTS code.

For data pre-processing, the Bol2017 data set was downloaded from the public data set collection (WEB, a) at 25.10.19.

After that first steps of pre-processing were performed as follows:

1. Filtering out only those records, where HTS is not null and selecting required fields: Item Description, HTS.
2. Removing any non-alphabetic characters from an item description and removing all records, where an item description is empty. This procedure may lead to duplicates in data as for example "laptop123" and "laptop321" give after cleaning process the similar result "laptop" (see for example Table 2).
3. Removing all records, where HS code is not numerical or less than 6 characters long or HS code chapter is out of bound.
4. Removing all records, such that the texts are the same, but HS codes are different, leaving only one such pair.

As a result of data pre-processing, we received approximately 1.2 million rows containing HS codes and descriptions of items. We used this data set for machine learning tasks.

### 6.2 Training Doc2Vec model

**6.2.1 Data preparation for training.** We used Gensim library (WEB, b) and Python programming language in order to implement our method. For training the Doc2Vec model we performed some data preparation steps and transformed texts to tagged documents as follows:

1. Transforming text to set of words, removing all the following words from the set $W = \{"hs", "hscode", "hts", "htscode", "tel", "code", "pcs", "kg"\}$ and common words (stop words), such as articles, "for", "and", etc. The exact list can be found in (Stone et al., 2011) and all words with length 2 or less.
2. Tagging documents with a unique tag (for differentiating them). We use as a tag the index number of the document in the data set.
3. Adding non-unique tags to documents according to the Gensim library (WEB, b). In our case, the HS code, which comes together with the product description, will be considered as the second tag.

The result of data preparation is a collection of tagged documents in the following form: `(words=["adapter", "laptop", "battery", "mouse", "docks"], tags=["1035312", "847180"])`. As we see from the example above, the unique tag for the document is 1035312 and the non-unique tag is the HS code 847180.

For testing our models, we use the train–test method that requires two data sets. We split a collection of tagged documents with proportion 90/10, where 90 percent of documents go as a training set and 10 percent as a testing set.

**6.2.2 Model initialization.** The initial learning rate – alpha is chosen to be 0.025 for Doc2Vec PV-DBOW implementation employed by us. We do not allow the learning rate to drop as training progresses (so, minimal alpha is the same). We take into vocabulary all words: so, we do not ignore words with a total frequency lower than some value. For training, we use PV-DBOW as training algorithm and the number of iterations (epochs) over the corpus is 10.

Doc2Vec is an unsupervised machine learning algorithm. Therefore, if only texts are used, then it is hard to estimate the correctness of the model. However, the manual inspection of the results is applicable.

### 6.3 Using Doc2Vec model

**6.3.1 Calculating text similarities of descriptions of goods.** The Doc2Vec model allows to find the most similar documents to the original text. Suppose we choose an original text as "laptop computer" and try to find the texts that are similar to the original one. First of all, we need to pre-process the original text and transform it into the vector using pre-trained Doc2Vec model. Secondly, we can find the closest texts, by measuring cosine vector similarities between the original text vector and texts in the vocabulary. As a result, we have cosine similarities between the given original text and the most similar documents. For example, in Table 2 (columns one and three) top 8 most similar documents to the original text "laptop computer" are provided. This result makes it possible to do some manual inspection in order to see the meaningfulness of provided similarities.

**6.3.2 HS code prediction.** We can apply the previously described text similarity approach for HS code prediction. As was mentioned in Section 6.2.1 we have a set of tagged documents and one of the tags is a HS code. We try to find the most similar documents to the original text document, for example, ”laptop computer”, but now we also take into account tags. Results including top 8 most similar documents to the text ”laptop computer” together with predicted HS codes (i.e. tags) are shown in Table 2. We see from the table that high text similarity is not always related to the same values or similar values of predicted HS codes (our first hypothesis does not always hold).

**Table 2.** Eight most similar documents to the original text document ”laptop computer” and their tags.

| $CosSim$ | Predicted HS code | Textual description |
|---|---|---|
| 0.886 | 847150 | desktop laptop |
| 0.886 | 847130 | laptop |
| 0.882 | 847130 | laptop desktop |
| 0.88 | 691390 | laptop sleevewacthband |
| 0.879 | 420212 | laptop bag laptop bag laptop |
| 0.873 | 847130 | laptop |
| 0.871 | 847330 | laptop stand |
| 0.864 | 420222 | laptop bag |

## 6.4 HS code recommendation based on text similarity

**6.4.1 Recommending several HS codes.** We can use the results of the most similar text search to recommend HS codes. Depending on how many similar texts we return we can recommend a different number of HS codes. For example, if we take the similar tagged texts from the previous example, we can recommend to original text, which is ”laptop computer”, the following codes: 847150, 847130, 691390, 420212, 847330, and 420222. In this experiment we returned 8 similar texts, but as a result, we have only 6 different codes because of repetitions.

It is important to estimate the quality of HS code recommendations. Table 3 presents the experimental results of one of the possible ways to evaluate recommendations. First of all, we choose the number of returned similar texts $N_{texts}$. Secondly, for every document in a sample set (samples of size $S = 5000$ and $S = 10000$) we find a vector representation for that document, return the required number of similar texts and infer the HS codes with respect to tags. Thirdly, we check if the correct HS code (which comes with test set document) is in the set of recommended codes. Let’s denote the percentage of occurrences of the true value HS code in predicted codes as $O_S$, where $S$ is a sample size. Table 3 shows the results. For example, if we return only 8 similar documents, for the sample with size 5000, in the 80 percent of the cases the true value HS code will be in the set of predicted codes, for the entire test set, this value is 79.6 percent.

**Table 3.** Evaluation of HS code recommendations.

| $N_{texts}$ | $O_{5000}$ | $O_{10000}$ |
|---|---|---|
| 7 | 79.5% | 79.1% |
| 8 | 80.2% | 79.8% |
| 10 | 81.6% | 81.0% |
| 20 | 85.3% | 84.6% |
| 50 | 88.9% | 88.2% |
| 100 | 90.9% | 90.5% |
| 200 | 92.8% | 92.4% |
| 500 | 94.8% | 94.7% |
| 1000 | 96.3% | 96.0% |

**6.4.2 Recommending the unique HS code.** Recommending only one HS code is a more complex task. We need two main components as follows: a list of similar texts with predicted HS codes returned by Doc2Vec model and the algorithm (method) for choosing only one of HS codes out of the set.

The first, the most obvious option for the algorithm is to use the mode function that returns the most frequently used value that will be the recommended code. However, if we have multi-mode the algorithm will return only one of them. For mode function, the weight of each occurrence is one. In contrast, we may evaluate each occurrence with some weight function. Therefore, we use a weighted mode function, where the weights are cosine similarities of documents. This approach can help to minimize the number of multi-mode cases and also takes into account distances between texts.

**Table 4.** Recommending the unique HS-code: accuracy and F-Score values.

| $N_{texts}$ | Accuracy $S = 5000$ | F-Score $S = 5000$ | Accuracy $S = 35000$ | F-Score $S = 35000$ |
|---|---|---|---|---|
| 7 | 0.619 | 0.672 | 0.612 | 0.617 |
| **8** | **0.622** | **0.675** | **0.613** | **0.618** |
| 9 | 0.614 | 0.667 | 0.608 | 0.613 |
| 100 | 0.501 | 0.565 | 0.500 | 0.509 |

For example, according to Table 2, the most frequent value of HS code is 847130. Using weighted mode function its weighted modal value is 2.641 (using mode function its frequency is 3). Table 4 shows the results of the experiment for the test samples with sizes 5000 and 35000. The same experiment fulfilled on the entire test set with $N_{texts} = 8$ gives accuracy 0.613 and F-score 0.609.

As our experiments on smaller sample sets have shown (see Tables 3 and 4) that adding more returned texts does not improve the results and the optimal value is 8 similar documents to be returned for the HS code correctness assessment. This optimal value is taken into account in finding semantic similarity and combined similarity as

well as in computing HS code assessment score. According to that, these calculations (see below) are based on top 8 similar documents returned by the Doc2Vec model.

### 6.5 Computing semantic and combined similarity

Semantic and combined similarities are calculated according to the formulas (in Equation 7) and (in Equation 11). As an example, based on the HS code taxonomy in Fig. 1 and cosine similarities of the given texts, combined similarities for some original HS codes compared to predicted HS codes are calculated and results are provided in Table 5.

Cosine similarities have been presented in Table 2. In Table 5 we use the average value of cosine similarities in the case if more than one similar HS codes are predicted. We see that combined similarity compensates shortage of using only cosine similarity of textual descriptions to decide about the correctness of the corresponding HS code as it takes into account the HS code structure i.e. its position in HS code taxonomy. For example, in the case of original HS code 420212 for what the $WPSim = 1$, combined similarity measure assigns the highest score however the other predicted HS codes, except code 420222, with a higher level of cosine similarity of the text gets the value of zero as they belong to other sections then chapter 42 belongs to. The HS code 420222 is the next candidate for the suitable HS code.

**Table 5.** Similarity measures.

| $C_1$ | $C_2$ | Textual descriptions for $C_2$ | $CosSim$ | $WPSim$ | $CombSim$ |
|---|---|---|---|---|---|
| 847130 | 847150 | desktop laptop | 0.886 | 0.75 | 0.665 |
| | 847130 | laptop/ laptop desktop | 0.88 | 1 | 0.880 |
| | 691390 | laptop sleevewacthband | 0.88 | 0 | 0 |
| | 420212 | laptop bag | 0.879 | 0 | 0 |
| | 847330 | laptop stand | 0.871 | 0.5 | 0.436 |
| | 420222 | laptop bag | 0.864 | 0 | 0 |
| 420212 | 847150 | desktop laptop | 0.886 | 0 | 0 |
| | 847130 | laptop/ laptop desktop | 0.88 | 0 | 0 |
| | 691390 | laptop sleevewacthband | 0.88 | 0 | 0 |
| | 420212 | laptop bag | 0.879 | 1 | 0.879 |
| | 847330 | laptop stand | 0.871 | 0 | 0 |
| | 420222 | laptop bag | 0.864 | 0.75 | 0.648 |
| 850440 | 847150 | desktop laptop | 0.886 | 0.25 | 0.222 |
| | 847130 | laptop/ laptop desktop | 0.88 | 0.25 | 0,220 |
| | 691390 | laptop sleevewacthband | 0.88 | 0 | 0 |
| | 420212 | laptop bag | 0.879 | 0 | 0 |
| | 847330 | laptop stand | 0.871 | 0.25 | 0.218 |
| | 420222 | laptop bag | 0.864 | 0 | 0 |

### 6.6 Computing and evaluating HS code assessment score

Computing HS codes assessment scores is performed using the method described in Section 5.3. For example, according to this method (see also data provided in Tables 1 and 5) the assessment score for the original HS code 847130 is 4 (the highest score), for code 420212 it is 4, and for code 850440 it is 1.

For the evaluation of HS code assessment scores obtained by our method, we set up the following experiment. The data is splitted for training and testing sets as in Section 6.2.1. The Doc2vec model is trained on training data. In the test set, we assign the $HSScore$ for each pair of a textual description and the corresponding (given) HS code. We assume that HS codes are correct for all data, so we expect that the most of pairs will be graded with the HS code assessment score of 4 or 3. According to experimental results (see Table 6), approximately 80 percent of cases the score was 4 or 3, which is a very good result and shows that our hypothesis holds true of 80 percent of cases. Consequently, our method provided in this paper works well.

Experimental results are given in Table 6. Where $HSScore$ denotes the HS code assessment score (Table 1) and $S_{5000}$, $S_{10000}$, $S_{full}$ – size of the test set.

**Table 6.** Experiment: Evaluation HS Code Assessment Score.

| $HSScore$ | $S_{5000}$ | $S_{10000}$ | $S_{full}$ |
|---|---|---|---|
| 0 | 262 (5.24%) | 572 (5.72%) | 6882 (5.88%) |
| 1 | 140 (2.8%) | 303 (3.03%) | 3462 (2.96%) |
| 2 | 347 (6.94%) | 688 (6.88%) | 8084 (6.91%) |
| 3 | 276 (5.52%) | 551 (5.51%) | 6417 (5.48%) |
| 4 | 3974 (79.48%) | 7885 (78.85%) | 92046 (78.74%) |

## 7 Conclusions and discussion

In order to automate a time demanding and an error-prone task of assessment of HS code correctness in electronic cross-border trade the paper provides a machine learning based solution. The novel method is based on a new original combined similarity measure of concepts that takes into account the semantic similarity of concepts in taxonomy as well as the cosine similarity of their corresponding textual descriptions. In the paper, maximality, positiveness and symmetry properties are proved for this new similarity measure. These properties are important for using our combined similarity measure also in many other similar applications.

In this paper, the Doc2Vec model was trained on product descriptions from real data set of Bill of Lading Summary 2017 and used for predicting HS codes on the basis of the cosine similarity of product descriptions. The HS code correctness assessment calculation uses a novel combined similarity measure based on cosine similarity of product descriptions and the Wu and Palmer semantic similarity of HS codes computed on the basis of the HS code taxonomy. As a result, we have shown that using the HS

code taxonomy as a complementary knowledge to pure textual descriptions of products compensates insufficient knowledge obtained from short texts and makes HS codes correctness assessment scores more correct compared to pure text-based approaches.

We need to note that the quality of product descriptions in the real data set was rather poor. Product descriptions were extremely short and noisy. In many cases, texts did not contain any product-related words or included additional irrelevant symbols. Therefore, when applying the method on a data set of better quality the results of machine learning part of our method can be improved.

Finally, the experiments presented in the paper have shown that our method assigns accurate HS code correctness assessment scores in approximately 80 percent of cases showing that the method provided in this paper works well.

Concerning the future work, there are other options of semantic similarity measures to be used instead of Wu and Palmer semantic similarity in our combined similarity measure. Of course, the replacing measure should fulfill all the properties of the normalized similarity measure.

In addition, other feature extraction methods than Doc2Vec need to be investigated as well as suitable classifiers for automating the HS code correctness assessment task.

## Acknowledgments

This research was partially supported by the Institutional Research Grant IUT33-13 of the Estonian Research Council.

## References

Ding, L., Fan, Z., Chen, D. (2015). Auto-categorization of HS code using background net approach. *Procedia Comput. Sci.*, 60, 1462–1471.

Euzenat, J., Shvaiko, P. (2013). Ontology matching. *Springer-Verlag Berlin Heidelberg*.

Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. (2003). KNN model-based approach in classification. *In: Proceedings of On The Move to Meaningful Internet Systems 2003*, 1532–1543.

Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). Applied Logistic Regression. *Wiley*

Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A. (2016). Reading Text in the Wild with Convolutional Neural Networks. *Int. J. Comput. Vis.*, 116, 1–20.

Kappler, H. (2011). Reversing the trend: low cost and low risk methods for assuring proper duty payments. *World Cust. J.*, 5(2), 109–122.

Larson, R. R. (2010). Introduction to Information Retrieval. *J. Am. Soc. Inf. Sci. Tec.*, vol. 61(4), 852–853.

Le, Q., Mikolov, T. (2014). Distributed representations of sentences and documents. *In: Xing, E.P., Jebara, T. (eds.) Proceedings of Machine Learning Research*, 31st International Conference on Machine Learning (June 2014, Beijng, China), vol. 32, no. 2, 1188–1196.

Lecun, Y. Bengio, Y., Hinton, G. (2015). Deep Learning. *Nature 521*, 436–444.

Li, G., Li, N. (2019). Customs classification for cross-border e-commerce based on text-image adaptive convolutional neural network. *Electron. Commer. Res.*, 19(4), 779–800.

Manevitz, L. M., Yousef, M., Cristianini, N., Shawe-Taylor, J., Williamson, B. (2001). One-Class SVMs for Document Classification. *J. Mach. Learn. Res.*, 2, 139–154.

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. *In: Proceedings of Workshop at ICLR*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *In:NIPS*.

Noormanshah, W.M.U., Nohuddin, P.N.E., Zainol, Z. (2018). Document Categorization Using Decision Tree: Preliminary Study. *Int. J. Eng. Technol.*, 73, 437–440.

Pennington, J., Socher, R., Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Rocchio, J. (1965). Relevance feedback in information retrieval.

Ruder, D. (2020). *Application of Machine Learning for Automated HS-6 Code Assignment*. MSc thesis, Tallinn University of Technology, Tallinn, Estonia.

Salton, S., Buckley, S. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, 24, 513–523.

Sanchez, D., Batet, M., Isern, D., Valls, A. (2012). Ontology-based semantic similarity: a new feature-based approach. *Expert Syst. Appl.*, 39, 7718–7728.

Shenoy, M., Shet, K.C., Acharya, U.D. (2012). A new similarity measure for taxonomy based on edge counting *CoRR*, abs/1211.4709.

Spichakova, M., Haav, H-M. (2020). Using machine learning for automated assessment of misclassification of goods for fraud detection, *In: Databases and Information Systems: 14th International Baltic Conference*, 144–158.

Stone, B., Dennis, S., Kwantes, P.J. (2011). Comparing methods for single paragraph similarity analysis, *Top. Cogn. Sci.*, 3(1), 92–122.

Turhan, B., Akar, G.B., Turhan, C., Yukse, C. (2015). Visual and textual feature fusion for automatic customs tariff classification. *In: 2015 IEEE International Conference on Information Reuse and Integration*, 76–81.

Wang, P., Xu, J., Xu, B., Liu C.L., Zhang, H., Wang, F., Hao, H. (2015). Semantic Clustering and Convolutional Neural Network for Short Text Categorization. *In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 352–357.

Wu, Z., Palmer, M. (1994). Verbs semantics and lexical selection. *In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 133–138.

Xu, B., Guo, X., Ye, Y., Cheng, J. (2012). An improved random forest classifier for text categorization. *J. Comput. (Finland)*, 7, 2913–2920.

Yang, K., Kim, W.J., Yang, J., Kim, Y. (2013). Ontology matching for recommendation of HS code. *In: HKICEAS-773*.

WEB (2017a). *European Court of Auditors: Special report no. 19, Import procedures (2017)*. `https://www.eca.europa.eu/en/Pages/DocItem.aspx?did=44169`

WEB (2017b). *EU VAT Reform (2017)*. `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017L2455&qid=1525703095429&from=EN/`

WEB (a). *Data set of Bill of Lading Summary 2017*. `https://public.enigma.com/datasets/bill-of-lading-summary-2017/0293cd20-8580-4d30-b173-2ac27952b74b`

WEB (b). *Gensim Python library*. `https://radimrehurek.com/gensim/`

WEB (c). *World Customs Organization (WCO), What is the Harmonized System (HS)?* `http://www.wcoomd.org/en/topics/nomenclature/overview/what-is-the-harmonized-system.aspx`

WEB (d). *Harmonized System Codes*. `https://tradecouncil.org/harmonized-system-codes/`

WEB (e). *HS Nomenclature 2017 edition*. `http://www.wcoomd.org/en/topics/nomenclature/instrument-and-tools/hs-nomenclature-2017-edition/hs-nomenclature-2017-edition.aspx`

WEB (f). *Semantics3. Categorization API*. https://www.semantics3.com/products/ai-apis#categorization-api
WEB (g). *3CE HS classification and verification solution*. https://www.3ce.com/solutions/

Received November 30, 2020 , accepted November 30, 2020