Using Web Data to derive the Economic Activity of Enterprises

Manveer Mangat, Statistics Austria

Abstract

The economic activity of enterprises (NACE) is often a key characteristic for the production of business statistics. The NACE code is determined for all units in the population and stored in the statistical business register. It is imperative that misclassifications in NACE codes are avoided, since they can lead to seriously biased business statistics. Determining and maintaining the main economic activity of an enterprises is a challenging classification task which relies on manual editing. Accordingly, this leads to the swift depletion of time resources, especially for national statistical institutes with a large business population size. Due to an increasing web presence of enterprises, web data is becoming a viable data source to help classify the economic activity of enterprises. One project within the ESSNet Web Intelligence Network, which started in April 2021, aims to develop automatic procedures to support manual editing of NACE codes. Clearly whether a proposed classification model has the ability to support the manual editing processes will depend on its quality. Hence the purpose of this paper is two-fold: 1) to construct a hierarchical classification model to predict NACE 1-5 codes of enterprises on the basis of their scraped websites and 2) to apply evaluation measures, including a novel customized performance measure, which are more suitable to assess the quality of hierarchical models than the standard evaluation metrics. Our web data encompasses the web pages of enterprises that were part of the Information and Communication Technologies survey from 2019 to 2021.

1 Introduction

NACE is the European standard hierarchical classification method used to classify enterprises according to their economic activity. As such it builds the foundation of various business statistics and indicators which in turn are the basis of policy proposals and implementations, hence underlining the importance of correct classifications. Any misclassification - in particular of larger enterprises - will inevitably lead to biased statistical outputs. Accordingly national statistical institutes carefully classify and edit NACE codes continuously to ensure a high level of accuracy which causes a significant depletion of time resources.

In order to assist and expedite the manual classification and editing process, we propose a hierarchical classification model that uses the scraped web pages of enterprises to predict their respective NACE level 1-5 codes. Clearly whether the proposed classification model will have the ability to support the manual classification and editing process will depend on its quality. Thereby the drawback of using the usual evaluation metrics such as the precision, recall and accuracy to evaluate hierarchical text classification models is their incapacity to account for the *relationship* or *closeness* between the categories in the hierarchy. Accordingly Sun and Lim (2001) proposed adjusted versions of the standard performance metrics which incorporate the information pertaining to class relationships in a hierarchical structure, to provide a more accurate evaluation method for hierarchical models.

The purpose of this paper is tow-fold: 1) to construct a hierarchical classification model based on the local classifier per parent node approach - which trains each parent node to only distinguish between its child nodes, hence respecting class consistency by construction - to predict NACE level

1-5 codes of enterprises on the basis of their scraped websites and 2) to evaluate it using the adjusted versions of the standard measures, as well our newly proposed performance measure, which are more suitable to asses the performance of hierarchical models than the standard evaluation metrics.

The paper is structured as follows. The second section will give a brief description of various quality metrics encompassing the adjusted versions of the standard evaluation measures which account for the degree of alignment between the true and predicted classes, as well as our newly proposed evaluation measure which weighs classes according to their respective sizes. Section three presents the methodological framework used to construct the hierarchical classification model along with the results of its application to the regarded web data set consisting of the web pages of enterprises that were part of the Information and Communication Technologies (ICT) survey in 2019, 2020 and 2021. Section 4 presents the evaluation results of the hierarchical model using the quality measures from Section 2 and Section 5 concludes.

2 Evaluation Measures

In this section we will present the adjusted standard evaluation measures based on *class similarity* and *class distance* (see Sun and Lim 2001), which account for the degree to which the true and predicted classes align. In addition we will also present extended versions of the standard evaluation measures specifically designed for hierarchical classification tasks. Finally we will propose an evaluation measure specifically designed for our use case. Henceforth the words class and category will be used interchangeably and we adopt the following notation: the category space is denoted by $\mathcal{C} = \{C_1, \dots, C_m\}$ where m is the depth of the tree (i.e. there are m levels) and for each level $l \in \{1, \dots, m\}$, $C_l = \{c_{l1}, c_{l2}, \dots, c_{ln_l}\}$ contains all the classes on the l-th level of the tree and n_l denotes its cardinality, i.e $|C_l| = n_l$. For the total number of classes across all levels we denote $n := |\mathcal{C}|$. (Note, for ease of notation and to increase readability, we will occasionally omit one subindex of classes $c_{li} \in C_l$ for $l \in \{1, \dots, m\}$ if convenient).

We define the dimension of the feature vectors for our model at level l to be $d_l := n_l f(l)$, where $f : \{1, \dots, m\} \longrightarrow A \subset \mathbb{N}$ is a function assigning the number of features that are considered for each class for a given level l, which implies that for each class on level l the same number of features will be considered.

2.1 Evaluation Measures based on Class Similarity

One way to compute the degree of alignment between two classes $c_{li} \in \mathcal{C}_l (\subset \mathcal{C})$ and $c_{lj} \in \mathcal{C}_l (\subset \mathcal{C})$ from the l-th level is to compute their *category similarity* $CS(c_{li}, c_{lj})$ using the cosine similarity

$$CS(c_{li}, c_{lj}) := \frac{\sum_{k=1}^{d_l} (f_k^{li} f_k^{lj})}{\sqrt{\sum_{k=1}^{d_l} (f_k^{li})^2} \sqrt{\sum_{k=1}^{d_l} (f_k^{lj})^2}}, \tag{1}$$

where $f^{li} \in \mathbb{R}^{d_l}$, $f^{lj} \in \mathbb{R}^{d_l}$ are the sum of all feature vectors assigned to class c_{li}, c_{lj} respectively. Using this category similarity measure we can define the *Average Category Similarity* (ACS) on the l-th level in the following way

$$ACS_l := \frac{2 \sum_{i=1}^{n_l} \sum_{j=i+1}^{n_l} CS(c_{li}, c_{lj})}{n_l (n_l - 1)}, \tag{2}$$

where n_l is the total number of classes on the l-the level. Note the category similarity is only computed for classes on the *same* level. Now we can compute the category similarity between the

true class y of an enterprise e and its predicted class p. If the enterprise e belongs to the class c_i and $p = c_i$ i.e. $e \in TP_i$, then e is counted as 1 in the computation of precision and recall for the class c_i. However if e does not belong to the class c_i but it is predicted to be, i.e. $e \in FP_i$, we compute the similarity between the predicted wrong class $p = c_i$ and the true class y to determine how much e can actually *contribute* to class c_i when its precision and recall are computed. If the predicted and true class are very similar, the enterprise e will be counted close to a value of 1 for TP_i. Similarly if e is wrongly rejected from the class c_i, i.e. $e \in FN_i$, we can compute the similarity between the predicted class p and the true class $y = c_i$ to determine the contribution of e to the class c_i. The contribution of the enterprise e to the class c_i when $e \in FP_i$ or $e \in FN_i$ is given as follows

$$Conb(e, c_i) = \begin{cases} min\Big(1, max\Big(-1, \frac{CS(y,c_i)-ACS}{1-ACS}\Big)\Big), & \text{where } e \in FP_i \\ min\Big(1, max\Big(-1, \frac{CS(p,c_i)-ACS}{1-ACS}\Big)\Big), & \text{where } e \in FN_i, \end{cases} \tag{3}$$

where the range has been restricted to $[-1, 1]$. For all enterprises e that were incorrectly assigned to c_i, i.e $e \in FP_i$, the total contribution of these enterprises to category c_i is given by

$$FpConb_i := \sum_{e \in FP_i} Conb(e, c_i) \tag{4}$$

and similarly for all enterprises e that were incorrectly rejected from c_i, i.e $e \in FN_i$ the total contribution of these enterprises to category c_i is given by

$$FnConb_i := \sum_{e \in FN_i} Conb(e, c_i). \tag{5}$$

Using the category similarity we can now define the adjusted evaluation measures of: 1) precision and recall (see Equations (6)), 2) micro-averages of precision and recall (see Equations (7)), 3) macro-averages of precision and recall (see Equations (8)), 4) accuracy and error (see Equations (9)), for every class $c_i \in \mathcal{C}$ in the following way:

$$PR_i^{CS} = \frac{max(0, TP_i + FpConb_i + FnConb_i)}{TP_i + FP_i + FnConb_i}, \quad RE_i^{CS} = \frac{max(0, TP_i + FpConb_i + FnConb_i)}{TP_i + FP_i + FpConb_i} \tag{6}$$

$$PR_{Mic}^{CS} = \frac{\sum_{i=1}^{n} max(0, TP_i + FpConb_i + FnConb_i)}{\sum_{i=1}^{n}(TP_i + FP_i + FnConb_i)}, \quad RE_{Mic}^{CS} = \frac{\sum_{i=1}^{n} max(0, TP_i + FpConb_i + FnConb_i)}{\sum_{i=1}^{n}(TP_i + FN_i + FpConb_i)} \tag{7}$$

$$PR_{Mac}^{CS} = \frac{\sum_{i=1}^{n} PR_i^{CS}}{n}, \quad RE_{Mac}^{CS} = \frac{\sum_{i=1}^{n} RE_i^{CS}}{n}, \tag{8}$$

$$AC_i^{CS} = \frac{TP_i + TN_i + FpConb_i + FnConb_i}{TP_i + TN_i + FP_i + FN_i}, \quad ER_i^{CS} = 1 - AC_i^{CS}, \tag{9}$$

where TN_i are all the observations that are correctly rejected from the class c_i.

2.2 Evaluation Measures based on Class Distance

Another way to quantify the degree of alignment between two classes $c_{li} \in \mathcal{C}_l$ and $c_{lj} \in \mathcal{C}_l$ from the l-th level is to compute their *distance* $Dis(c_{li}, c_{lj})$, which is defined as the total number of edges in the shortest path from c_{lj} to c_{lj}. The shorter the distance between two classes the higher the degree of alignment. As we will construct our hierarchical classification model using the local classifier per parent node approach, we can systematically derive the distance between two classes. Under the local classifier per parent node approach a false prediction made on level $l \in \{1, \ldots, m-1\}$

will result in false predictions on the subsequent level(s) $l+1$ as well. Indeed, if the hierarchical predictions p_i of an enterprise align with the true values y_i up until the l-th level i.e. $p_i = y_i$ for $i = 1, \ldots, l$ but do not coincide on the $(l+1)$-st level - hence also not on the subsequent levels, i.e. $p_i \neq y_i$ for $i = l+1, \ldots, m$, then the distance between the true category and the predicted category on the i-th level is given by $(i-l)2$, where $i = l+1, \ldots, m$. Before defining the contribution of an enterprise e to a class $c_i \in \mathcal{C}$, we have to specify an acceptable distance Dis_θ between two classes. For example if $Dis_\theta = 2$, then classes that are more than two edges apart will have a negative contribution to class c_i. A common way of selecting Dis_θ is to use the depth m of the tree. Analogous to Section 2.1, we can now define the contribution of the enterprise e to the class c_i, when $e \in FP_i$ or $e \in FN_i$ is given, as follows

$$Conb(e, c_i) = \begin{cases} min\Big(1, max\Big(-1, 1 - \frac{Dis(y,c_i)}{Dis_\theta}\Big)\Big), & \text{where } e \in FP_i \\ \Big(1, max\Big(-1, 1 - \frac{Dis(p,c_i)}{Dis_\theta}\Big)\Big), & \text{where } e \in FN_i, \end{cases} \tag{10}$$

where the range has been restricted to $[-1, 1]$. Similarly to Section 2.1, we can now define the adjusted evaluation measures of: 1) precision and recall 2) micro-averages of precision and recall 3) macro-averages of precision and recall 4) accuracy and error for every class $c_i \in \mathcal{C}$, by replacing the contribution by the class similarity, by that of the category distance in equations (6)-(9).

2.3 Hierarchy based Evaluation Measures

Kiritchenko et al. (2006) proposed a hierarchy based extension of the standard evaluation metrics precision hPR and recall hRE, defined in the following way

$$hPR := \frac{\sum_{i=1}^{K} |P_i \cap Y_i|}{\sum_{i=1}^{K} |P_i|} \quad hRE := \frac{\sum_{i=1}^{k} |P_i \cap Y_i|}{\sum_{i=1}^{K} |Y_i|}, \tag{11}$$

where K is the number of observations in the test set, P_i is the set consisting of the most specific class prediction(s) of the observation i i.e. $P_i = \{p_1^i, \ldots, p_{\tilde{m}}^i\}$ where $\tilde{m} \leq m$, (m being the number of levels) and Y_i is the set consisting of the true classes of the observation i on *every* level i.e. $Y_i = \{y_1^i, y_2^i, \ldots, y_m^i\}$.

2.4 Tailored Evaluation Measures

The measures considered so far, all gave equal weights to each class when computing the overall (average) quality of the model. However for our particular use case it is advisable to give more weight to classes to which large enterprises tend to be assigned, as a missclassified large enterprise will introduce a more significant bias in a business statistic than a small missclassified enterprise. As our statistical business registry does not provide the turnover of the enterprises, we use the number of employees of the enterprises as an indicator for their size as it is generally justifiable to assume that enterprises with a large number of employees will comparatively have a bigger turnover than enterprises with a smaller number of employees.

Accordingly we introduce the following average weighted precision PR_l, recall RE_l and accuracy AC_l for each NACE level $l \in \{1, \ldots, m = 5\}$, where we weight each class $c_{li} \in C_l = \{c_{l1}, c_{l2}, \ldots, c_{ln_l}\}$ with the total number of employees s_{li} of the enterprises that are assigned to this class

$$PR_l := \frac{1}{\sum_{i=1}^{n_l} s_{li}} \sum_{i=1}^{n_l} s_{li} PR_{li}, \quad RE_l := \frac{1}{\sum_{i=1}^{n_l} s_{li}} \sum_{i=1}^{n_l} s_{li} RE_{li}, \quad AC_l := \frac{1}{\sum_{i=1}^{n_l} s_{li}} \sum_{i=1}^{n_l} s_{li} AC_{li}, \tag{12}$$

Note these measures can be computed by either using the standard precision, recall and accuracy measures or their adjusted versions based on category distance and category similarity. To obtain an average value for the precision, recall and accuracy across all levels, we compute the (weighted) average of PR_l, RE_l and AC_l, respectively:

$$PR^w := \frac{1}{\sum_{l=1}^{m} |w_l|} \sum_{l=1}^{m} |w_l| PR_l, \quad RE^w := \frac{1}{\sum_{l=1}^{m} |w_l|} \sum_{l=1}^{m} |w_l| RE_l, \quad AC^w := \frac{1}{\sum_{l=1}^{m} |w_l|} \sum_{l=1}^{m} |w_l| AC_l, \quad (13)$$

where $m = 5$ is the total number of levels and $|w_l|$ refers to the number of predictions available at the l-th level. Note w_1 by construction will always be equal to the maximum number of predictions possible, i.e. will coincide with the number of enterprises contained in the test set and for the second NACE level we expect a similar result. However for NACE codes beyond the third level, the available number of predictions will tend to diminish for the local classifier per parent node approach as there might not be sufficient data to train the local classifiers.

3 Methodology

In this section the methodological framework used to construct the hierarchical classification model to predict the NACE level 1-5 codes of enterprises, based on their respective web scraped data, is presented.

3.1 Data

Before an enterprise can be classified on the basis of its webpage (URL), it first has to be linked to its true URL. For our data set we consider all enterprises that were part of the Information and Communication Technologies (ICT) survey in 2019, 2020 and 2021. Our method of linking each of these enterprises to their respective websites using the Statistical Business Registry (SBR) (contains information of enterprises e.g. name, address, NACE codes, etc) is conducted in the following way: We first search the name and address of each enterprise which has a website according to the ICT, through the Google Search API and retain the first ten URLs as possible website candidates of the enterprise. Subsequently using the statistical software R (R Core Team 2023) and Selenium we scrape each website and use that data to conduct our URL linking procedure. In particular we look for direct identifiers such as the value added tax (VAT) or the company ID (CID) of the considered enterprise on each of the websites. If a match can be found the respective website(s) are linked to the enterprise. Through this linking procedure we could link 72,498 enterprises with their respective websites, hence providing us with 72,498 labeled observations for the construction of our training and test set.

3.2 Pre-Processing

Before the set of raw scraped webpages can be used for the classification task, it has to be thoroughly processed. From our scraped webpages we only retain the text elements and discard the html code. Thereby in addition to using the text elements of the scraped landing page, we also include the text elements of certain sub-pages that contain keywords in the link which suggest that they might harbor information pertaining to NACE code classifications (e.g. "enterprise", "company", "unternehmen", "about us", "über uns", etc.). Once the text of interest from each webpage is obtained, it is processed in the following way: 1) each word is transformed with the German morphological lexicon 2) all digits and punctuations are removed 3) all characters not part of the

German dictionary are removed 4) German stop words are removed and 5) lemmatization using the German version of the hunspell dictionary is conducted.

3.3 Feature Selection

After applying the pre-processing procedure to our data set, the number of total unique words across all the scraped webpages amounts to 3,212,247, hence providing a vast pool of words (features) to select form. The feature selection method we apply was proposed by Uysal (2016) where a global and a local feature selection score function is combined to select a balanced set of features for each class. We use the Distinguishing Feature Selector (DFS) as our global selection score and the Odds Ratio (OR) as our local selection score. As we use the local classifier per parent node approach, where each parent node is only trained to distinguish between its children, we construct a multi-step feature selection procedure which respects this property. Let us first introduce some notation. Let L_l denote the set of all NACE level l codes occurring in the regarded data set where $l = 1, \ldots, 5 = m$ indicates the level. Accordingly $|L_l|$ denotes the number of all NACE level l codes. For our hierarchical classification method we will be working with restricted data sets such as $L_l^j = \{i \in L_l | parent(i) = j\}$, which contain all the children of the parent $j \in L_{l-1}$ at level l. For ease of notation we assign a number from 1 to $|L_l^j|$ to each class in L_l^j, then we can denote the feature set corresponding to L_l^j by $\mathcal{F}_l^j = \{F_{l1}, \ldots, F_{l|L_l^j|}\}$, where F_{lk} is the feature set of the k-th class in L_l^j. However the feature set is *not* computed for every class on every level. Indeed, the computation of the feature set for a class depends on two conditions:

Selection Conditions: Let $j \in L_l$ for $l = l, \ldots, 4$ be the parent on the l-th level, then the feature set $F_{(l+1)k}$ of the child $c_{(l+1)k} \in L_{l+1}^j$ is computed, where $k = 1, \ldots, |L_{l+1}^j|$, if

1. there are at least two children $|L_{l+1}^j| \geq 2$ of the parent j, i.e. there are at least two classes to be differentiated amongst
2. there are at least two enterprises that can be assigned to each class in L_{l+1}^j.

If we would not impose the first condition, then in case of a single child $c \in L_{l+1}^j$ of the parent j, the model would inevitably predict the NACE level $l+1$ code c for every enterprise in the test set with NACE level l code j. The second condition ensures that there is sufficient data available to compute useful features, as a single web page will most likely not provide a representative feature set of the respective class.

3.4 Construction of Training and Test Set

Our data set consists of enterprises that can be assigned to 19 out of the 21 possible NACE level 1 codes, i.e. $L_1 = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S\}$. As our aim is to construct a five tier hierarchical classification model, it is not only imperative to ensure that there are sufficient enterprises assigned to each of the occurring classes, but also that there are multiple children of a parent (see *Selection Conditions*). Accordingly we restrict our data set to all enterprises that are assigned to the NACE level 1 code $j \in L_1 := \{C, F, G, H, N\}$ to minimize the violations to the *Selection Conditions*. This restriction reduces the number of labeled data 72,498 to 45,105. Once the data set for the model construction is established, a representation method for the categorical features (words) has to be selected. We use the one-hot-encoding method to represent the categorical features as multidimensional vectors where each dimension corresponds to a feature which

we then weight by the term-frequency inverse document frequency. Hence the feature vector of an enterprise at the NACE level 1 - 5, is represented by a $f(1)|L_1| = 500|L_1|$, $f(2)|L_2^j| = 200|L_2^j|$ where $j \in L_1$, $f(3)|L_3^j| = 100|L_3^j|$ where $j \in L_2$, $f(4)|L_4^j| = 80|L_4^j|$ where $j \in L_3$ and $f(5)|L_5^j| = 50|L_5^j|$ where $j \in L_4$, dimensional feature vector, respectively where j denotes the parent. We then use 80% of our data as the training set (36,076 enterprises) and the remaining 20% as the testing set (9,060 enterprises). However, if we only retain the enterprises whose number of employees are available in the SBR - which is required for the computation of the tailored evaluation measure (see Section 2.4)- the number of test examples is reduced to 8805.

Note, we could have also used the whole data set to build the hierarchical classification model, but as we would like to ideally obtain a prediction at all the five NACE levels, we omit the enterprises for which we know with certainty that the model would have not been able to predict the NACE code on all the levels.

3.5 Model Specification

For our hierarchical classification model we use the XGBoost algorithm (see Chen and Guestrin 2016), which we implement using the R-package xgboost by Chen et al. (2015). Before an XGboost algorithm can be implemented its hyperparameters have to be selected. In order choose the hyperparamters we apply the random search technique, which generates random values for each hyperparameter and then uses a 5 cross-validation to find the optimum values.

Our hierarchical classification model consists of multiple mulit-class models which are trained in the following way: The first model is trained to distinguish between the classes of L_1. Subsequently we train a model for each class $j \in L_l$ on any of the levels $l = 1, \ldots, 4$, to differentiate among its children as long as 1) there are at least two children $|L_{l+1}^j| \geq 2$ of the regarded parent j and 2) there are at least two enterprises that are assigned to each of the children in L_{l+1}^j. Hence the *Selection Conditions* dictate the number of models that can be constructed on each level. The feature set used to train the first model is given $\mathcal{F}_1$ and the feature set used to train the rest of the models is given by $\mathcal{F}_{l+1}^j$. Clearly the data set used to train the latter models is restricted to the enterprises that are assigned to the NACE level l code j.

As mentioned, for a model to be trained restricted to a class the *Selection Conditions* have to be satisfied. If one of these conditions is violated the model is not trained for that particular class. In case of the hierarchical classification model this means that the hierarchical prediction will be terminated before the final level is reached.

4 Evaluation

Table 1 shows the performance results of the hierarchical classification model at NACE level 1. The performance was evaluated using the standard evaluation measures precision, recall and accuracy and their adjusted counterparts based on category similarity (see Section 2.1) and category distance (see Section 2.2). For the category distance we used the depth of our tree as the acceptable distance $D_\theta = 5$ which implies that only nodes that are more than 5 edges apart will contribute negatively to the class in question. As is apparent the adjusted versions of the standard evaluation are evermore higher than their standard counterparts, regardless of whether the category distance or category similarity is used to quantify the alignment between classes. The performance results of the model at NACE level 2-5 lead to the same conclusion (their tables have been omitted due to their significant size).

To obtain an overall value for the precision, recall and accuracy we consider the Micro-Averages

NACE	PR	RE	AC	PR^{CS}	RE^{CS}	AC^{CS}	PR^{CD}	RE^{CD}	AC^{CD}
C	0.818	0.866	0.901	0.884	0.922	0.936	0.932	0.952	0.960
F	0.872	0.861	0.953	0.857	0.848	0.949	0.953	0.948	0.981
G	0.850	0.851	0.903	0.901	0.911	0.936	0.945	0.945	0.961
H	0.911	0.851	0.982	0.901	0.833	0.980	0.967	0.943	0.993
N	0.905	0.821	0.969	0.914	0.834	0.971	0.966	0.932	0.988

Table 1: Performance results restricted to the first NACE level

Metric	Measure	Micro-Average	Macro-Average
-	Pr	0.74	0.71
-	Re	0.76	0.63
CS	Pr	0.84	0.81
CS	Re	0.86	0.74
CD	Pr	0.82	0.72
CD	Re	0.85	0.67

Table 2: Performance results over the whole category space. *Metric* describes if and which method (category similarity CS or category distance CD) was implemented to quantify the alignment between the predicted and true classes

and Macro-Averages which we define on the basis of the standard and adjusted evaluation measures (see Table 2). The Micro-Averages and Macro-Averages computed using the adjusted standard evaluation metrics are higher than their counterparts based on the standard evaluation metrics. Furthermore the hierarchical based precision is $hP = 0.74$ hence similar to the value obtained by computing the Micro-Average and Macro-Average using the standard evaluation metrics. The low value of the hierarchical recall $hR = 0.59$ can be explained by the unavailability of predictions at the more granular level, in particular the fourth and fifth NACE level (see Figure 2).

Table 3 displays the weighted precision at each separate NACE level on the basis of the standard evaluation metrics (PR,RE,AC) and the adjusted evaluation measures based on category similarity (PR^{CS}, RE^{CS}, AC^{CS}) and category distance (PR^{CD}, RE^{CD}, AC^{CD}) and its overall weighted average across all the five NACE levels and shows that it decrease as the NACE level increases. This is due to the fact that the enterprises for which the predictions are available on the more granular level are rather on the smaller side (do not have a large number of employees) and not due to a overall decreasing quality of performance as the level increases as Table 4 illustrates. Indeed the average performance over each NACE level is stable over time.

In order to illustrate the discrepancy between the predicted and true NACE codes at all levels for a fixed NACE level 1 code $j \in L_1$, we plot the distribution of all NACE codes at level 1-5 with NACE level 1 code $j = H$ from the test set (see Figure 1) and the distribution of all NACE codes at level 1-5 with NACE level 1 code H obtained from the hierarchical prediction model (see Figure 2). The class H was merely chosen due to its comparatively modest number of classes at each level which allows for a interpretable plot.

In Figure 1 all the enterprises with NACE level 1 H get partitioned into five distinct classes $H49$, $H50$, $H51$, $H52$, $H53$ with cardinality 503, 5, 10, 154, 23, respectively at the NACE level 2 corresponding to the colors blue, green, orange, violet, red respectively. Similarly each of these NACE 2 level codes get partitioned into multiple classes at the NACE level 3 themselves represented by the different shades of their respective colors. In particular the classes at level 3 consist of $H491$,

Metric	NACE 1	NACE 2	NACE 3	NACE 4	NACE 5	Weighted Average
PR	0.86	0.78	0.70	0.46	0.14	0.68
PR^{CS}	0.89	0.87	0.83	0.56	0.15	0.77
PR^{CD}	0.95	0.91	0.79	0.45	0.12	0.76
RE	0.85	0.74	0.68	0.47	0.16	0.67
RE^{CS}	0.88	0.83	0.80	0.56	0.17	0.75
RE^{CD}	0.94	0.90	0.77	0.46	0.16	0.75
AC	0.93	0.97	0.96	0.69	0.18	0.86
AC^{CS}	0.95	0.98	0.97	0.69	0.18	0.87
AC^{CD}	0.97	0.99	0.97	0.69	0.18	0.88

Table 3: Performance results in terms of weighted precision, recall and accuracy at 1) each level separately (NACE level 1- 5) and 2) over the whole category space (Weighted Average) using different evaluation metrics. *Metric* describes if and which method (category similarity or category distance) was implemented to quantify the alignment between the predicted and true classes

Metric	NACE 1	NACE 2	NACE 3	NACE 4	NACE 5	Average
PR	0.87	0.77	0.72	0.68	0.73	0.75
PR^{CS}	0.89	0.84	0.83	0.79	0.80	0.83
PR^{CD}	0.95	0.91	0.82	0.66	0.61	0.79
RE	0.85	0.67	0.58	0.59	0.83	0.70
RE^{CS}	0.87	0.77	0.71	0.72	0.90	0.79
RE^{CD}	0.94	0.86	0.70	0.58	0.77	0.77
AC	0.94	0.99	0.99	1.00	0.99	0.98
AC^{CS}	0.95	0.99	1.00	1.00	1.00	0.99
AC^{CD}	0.98	0.99	1.00	1.00	0.99	0.99

Table 4: Performance results in terms of average precision, recall and accuracy 1) at each level separately (NACE 1- 5) and 2) over the whole category space (Average) using different evaluation metrics. *Metric* describes if and which method (category similarity or category distance) was implemented to quantify the alignment between the predicted and true classes

$H492$, $H493$, $H494$, $H503$, $H511$, $H521$, $H522$, $H531$, $H532$ with cardinality 1, 5, 211, 286, 5, 10, 8, 146, 2, 21, repectively. Then each of the classes at the NACE 3 get partitioned into multiple classes at the NACE level 4 and each of the NACE level 4 codes get partitioned into multiple classes at the NACE 5 with the respective colors. Figure 2 illustrates the distribution of all the predictions with NACE level 1 code H. At the NACE level 2 the 5 distinct classes $H49$, $H50$, $H51$, $H52$, $H53$ are predicted with cardinality 518, 4, 6, 123 10, corresponding to the colors blue, green, orange, violet, red respectively just as in Figure 1. On level 3 the predicted classes are $H491$, $H492$, $H493$, $H494$, $H511$, $H522$, $H531$, $H532$ with the cardinality 1, 3, 214, 300, 6, 123, 2, 8, respectively. Recall that the hierarchical model might not provide a prediction beyond a certain NACE level in case one of the *Selection Conditions* is not satisfied. An unavailable prediction is indicated by opacity in Figure 2. For example the color green corresponds to the NACE level 2 code $H50$, however the opaque green color at NACE level 3 signifies that its prediction is not available at the third level, and hence by implication also not on the fourth and fifth NACE level. Generally no predictions are made at the fifth NACE level.

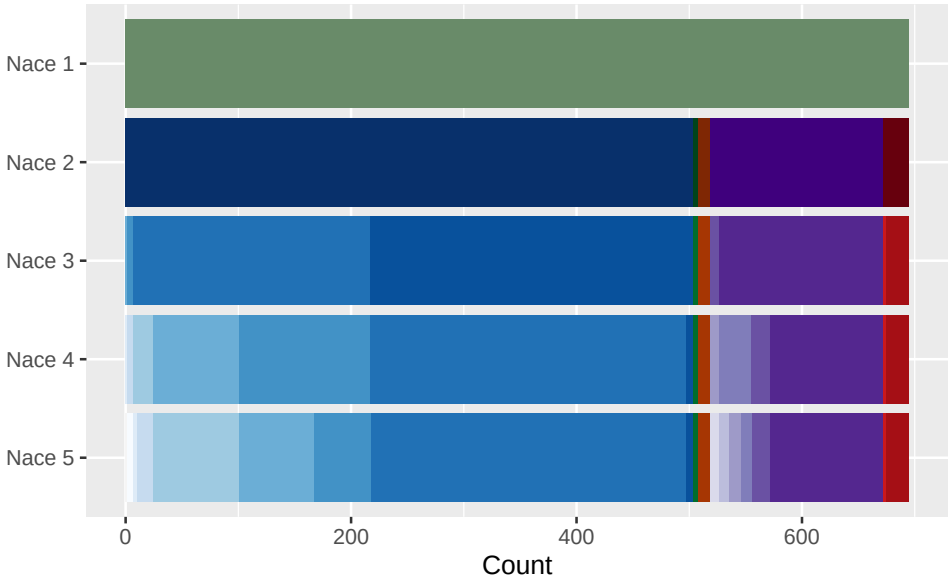

Figure 1: Distribution of the classes with NACE level 1 code H in the test set. The different colors correspond to the different classes. Thereby different shades of the same color indicate that they have the same parent at level 2.

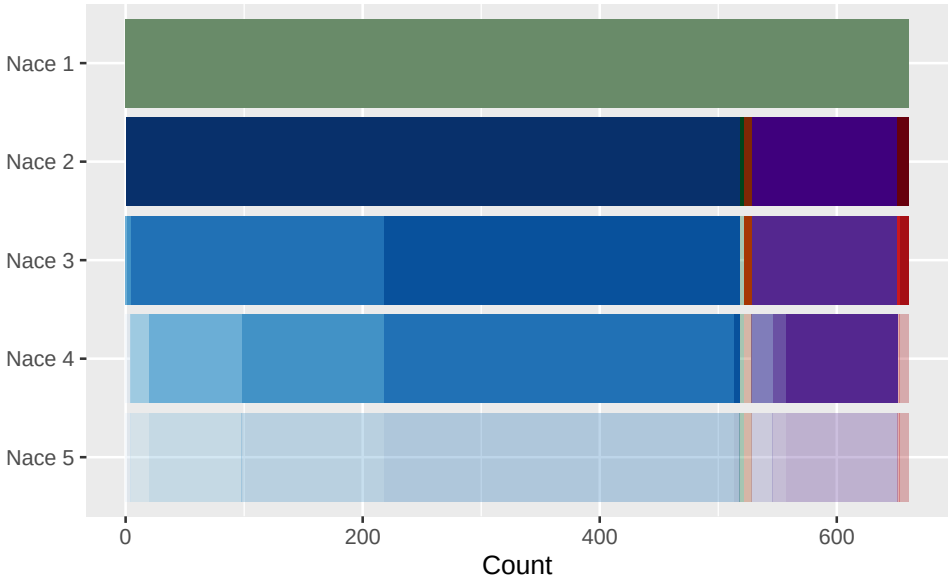

Figure 2: Distribution of the classes that were predicted to have NACE level 1 code H. The different colors correspond to the different classes. Thereby different shades of the same color indicate that they have the same parent at level 2. Opacity is indicative of no available prediction.

5 Conclusion

In this paper we construct a hierarchical classification model that predicts the NACE level 1-5 codes of enterprises on the basis of their scraped web pages and then we evaluate it using measures that are more suitable to asses the performance of hierarchical structures than the standard evaluation

metrics. We first give an overview of these suitable evaluation measures and then we propose a new evaluation measure specifically for our regarded use case, which weighs each class according to the number of employees that are contained in the considered class, allowing to give more weight to classes that tend to be assigned to large enterprises. Using the local classifier per parent node approach, we construct a hierarchical classification model by 1) pre-processing the web scraped data 2) computing the feature set for all classes of all NACE 1-5 levels by combing a global and a local feature selection score function as long as i. the class has at least one sibling and ii. there are at least two enterprises that can be assigned to each of the siblings 3) computing the feature vector for each enterprise using the one-hot-encoding method weighted by the term-frequency inverse document frequency 4) constructing a multi-class classifier using the XGBoost algorithm for each parent node as long as i. the parent node has at least two children and ii. there are at least two enterprises that can be assigned to each of the child using 80% of the data as the training set. Evaluating the model shows that the adjusted standard evaluation metrics, which account for the relationship between the predicted and true NACE codes of an enterprise, are higher than their standard counterparts. Accordingly, the overall performance of the model is deemed higher when the averages are computed using the adjusted evaluation metrics. Thereby the general performance of the hierarchical model measured in terms of the Micro-Average, Macro-Average and the tailored performance measure is of homogeneous nature.

References

Chen, Tianqi, and Carlos Guestrin. 2016. "Xgboost: A scalable tree boosting system." In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining,* 785–794.

Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. 2015. "Xgboost: extreme gradient boosting." *R package version 0.4-2* 1 (4): 1–4.

Kiritchenko, Svetlana, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. "Learning and evaluation in the presence of class hierarchies: Application to text categorization." In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19,* 395–406. Springer.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. https://www.R-project.org/.

Sun, Aixin, and Ee-Peng Lim. 2001. "Hierarchical text classification and evaluation." In *Proceedings 2001 IEEE International Conference on Data Mining,* 521–528. IEEE.

Uysal, Alper Kursat. 2016. "An improved global feature selection scheme for text classification." *Expert systems with Applications* 43:82–92.