Norwegian University
of Life Sciences

Master's Thesis 2023 60 ECTS
Faculty of Chemistry, Biotechnology and Food Science

Maximum Entropy COICOP Classification using Entity Forest

Louise R. Bauer-Nilsen
Bioinformatics & Applied Statistics

Acknowledgements

I would like to extend my gratitude to Statistics Norway for allowing me to collaborate with them on this thesis. I appreciate the trust they have shown throughout the research process and the resources they have shared with me.

I am also deeply grateful to my outstanding supervisors Li-Chun Zhang and Kathrine Frey Frøslie. Li-Chun Zhang for his unwavering guidance and support. His knowledge, expertise, and commitment have been priceless when shaping the direction of this thesis. And to Kathrine Frey Frøslie for her constant enthusiastic support and feedback.

Norwegian University of Life Sciences
Ås, 14th of May 2023

Louise R. Bauer-Nilsen

Abstract

This thesis proposes a generative approach to COICOP classification using entity resolution and maximum entropy classification as a formal framework. The current limitations in COICOP classification are related to the corpus of item descriptions and lack of data. I propose a new perspective on the classification task at hand, as I argue that the underlying problem in classification is the data itself. Therefore, corpus and feature engineering are crucial when improving classification. The proposed approach aims to engineer the corpus to construct an entity forest from the item descriptions, where terms in the description are mapped to the roots and branches of trees in the entity forest. The results of the proposed approach are illustrated by a proof-of-concept with data from Statistics Norway. This thesis provides insight into the problems with previous approaches to COICOP classification and shows how we potentially can achieve true resolution and more accurate classification.

Sammendrag

I oppgaven foreslår jeg en generativ tilnærming til COICOP klassifisering ved å se på entity resolution og maximum entropy klassifisering som en formell ramme. De nåværende begrensningene i COICOP-klassifisering er knyttet til varebeskrivelser og mangel på data. Jeg foreslår å se på klassifisering fra et nytt perspektiv og argumenterer for at det underliggende problemet i klassifiseringen er dataene selvslik at databehandling er avgjørende for å forbedre klassifiseringen. Den foreslåtte tilnærmingen har som mål å konstruere en entitetsskog fra varebeskrivelsene, der ord i beskrivelsen blir til røttene og grenene for trærne i skogen. Resultatene fra denne tilnærmingen illustreres med data fra Statistisk sentralbyrå. Denne oppgaven gir innsikt i problemene med eksisterende tilnærminger til COICOP-klassifisering og viser hvordan vi potensielt kan oppnå en bedre klassifisering.

Contents

List of Figures

List of Tables

1 Introduction

Statistics Norway (SSB) is the national statistical institute of Norway and the country's main provider of official statistics. SSB collects and produces statistics on the economy, population, and society in general. They also coordinate the statistics the Norwegian government produces and are responsible for Norway's participation in international statistics cooperation [SSB, 2022a]. Among the key statistics produced by SSB, we have *Household statistics*. Household statistics include data on Norwegian households' consumption and expenditure patterns and are collected in the household budget surveys (HBS) [Eurostat].

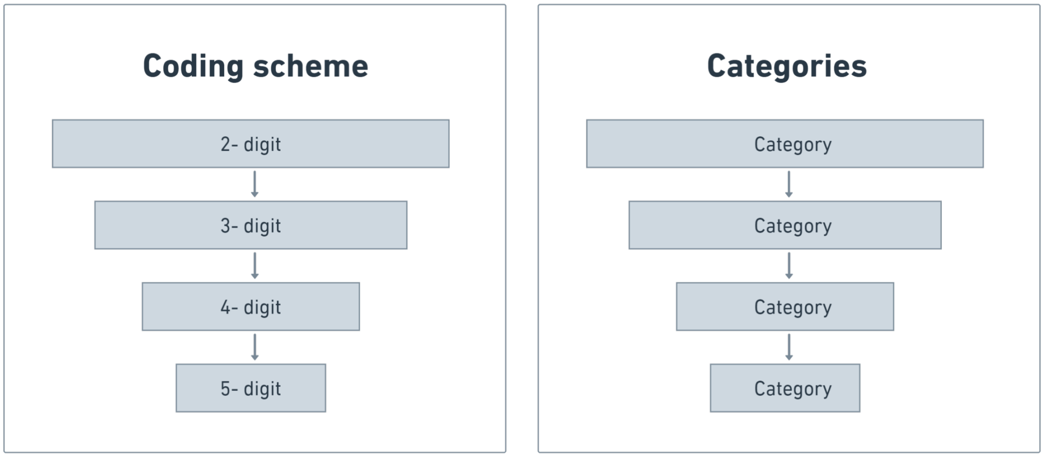
One of the main purposes of the HBS is to provide a detailed overview of private households' consumption of various goods and services to contribute to the measurement of social development and economic living conditions. The HBS is carried out in almost every country in the world and is crucial for calculating a country's Gross Domestic Product (GDP) and Price indices [Benedikt, Joshi, Nolan, Wolf, and Schouten, 2020]. To provide a detailed overview of private households' consumption and expenditure, a classification system, COICOP (Classification of Individual Consumption According to Purpose) was developed by the United Nations Statistics Division (UNSD). SSB has since 1974 carried out annual consumer expenditure surveys and In 1996, SSB introduced the classification of consumption, COICOP, for use in Norwegian statistics [SSB, 2022b]. The sample for the survey is drawn from individuals aged 0 to 79 (0-84 after 2012) and consists of personal interviews, receipts, and detailed accounting records from a representative sample of private households. In 2012, the sample size was 7,000 households. Before 2012 the sample size was 2,200 individuals annually [Holmøy and Lillegård, 2014].

1.1 COICOP Classification

Classifying household consumption expenditure at international level started in 1923. Three years later they had the first broad classification of household expenditure elaborated by the International Conference of Labour Statisticians. The first classification under the name COICOP was adopted by the United Nations Statistical Commission in March 1999

[UN, 2018]. Today’s classification is the outcome of a revision process that started formally in 2015 and involved contributions from numerous classification experts and users worldwide. COICOP 2018 is even more detailed than the previous versions, responding to the users’ need for more detail, addressing changes in goods and services within certain areas, improving the links of COICOP to other classifications, and addressing several international organizations’ emerging statistical and policy needs. In March 2018, the 49th Session of the United Nations Statistical Commission considered and endorsed the revised COICOP classification, COICOP 2018, as the internationally accepted standard [UN, 2018]

The COICOP 2018 follows a hierarchical structure consisting of four levels. The level of detail increases from a two-digit level to a five-digit level [UN, 2018]. The illustration given in Figure 1 demonstrates the coding system.

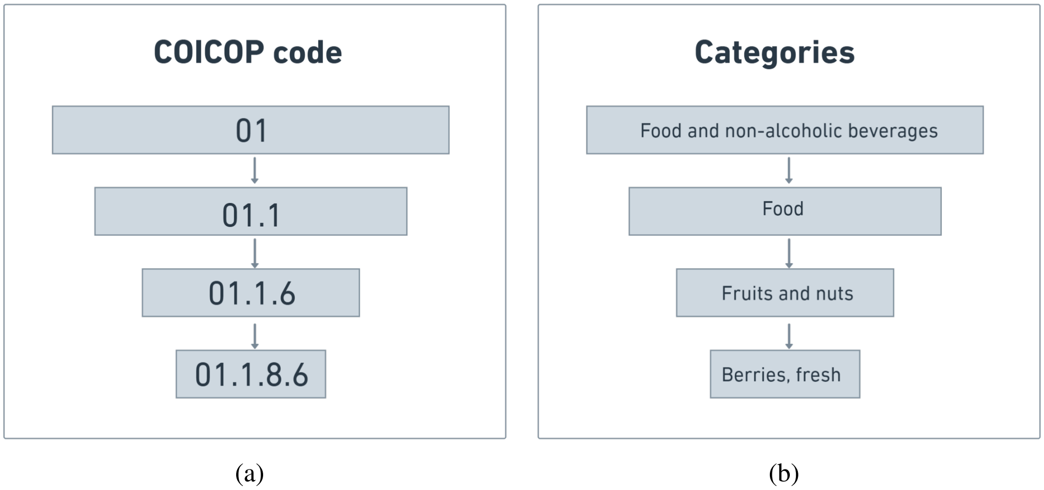


(a)

(b)

Figure 1: A visual representation of the hierarchical structure of the COICOP classification.

This means that the most detailed classification of an item happens in the 5-digit COICOP code. For example, an item description ’fresh strawberries’ belongs to the 5-digit COICOP code ’01.1.8.6’ named ’Berries, fresh’. The corresponding hierarchy is shown in Figure 2.



(a)

(b)

Figure 2: COICOP 2018 classification of 'Berries, fresh'.

1.2 Automatic COICOP Classification: Previous Work

Some recent studies have involved machine learning techniques for classifying items to the 5-digit COICOP code. In Benedikt et al. [2020] they present findings from research conducted as part of Work package 4.2 of the @HBS Project, focusing on automating the process of shopping receipts and classifying products to 5-digit COICOP codes using machine learning. The proposed end-to-end automation pipeline consists of modules for receipt scanning, image processing, optical character recognition, natural language processing, and machine learning classification. However, the experiments show that achieving pure automation while maintaining the required data quality for official statistics is challenging. While AI models can achieve up to 80 % accuracy, the last 20 % becomes increasingly difficult and expensive to achieve as algorithms become more complex. To mitigate this, the concept of Human-in-the-Loop is proposed, where machine and human intelligence combine to save time and resources on repetitive tasks while allowing humans to focus on value-added tasks that require flexibility and intelligence. The results from this paper coincide with the results made by Müller [2021], where his findings are consistent with results in Benedikt et al. [2020] approaching an accuracy score of 81% on the held-out test data in his study. Both studies have used their own data sets, so making any direct comparison with the results in the studies is challenging; however, the similar findings suggest that the models' suitability for 5-digit COICOP classification tasks may not be limited to a specific data set, as they used similar machine learning methods.

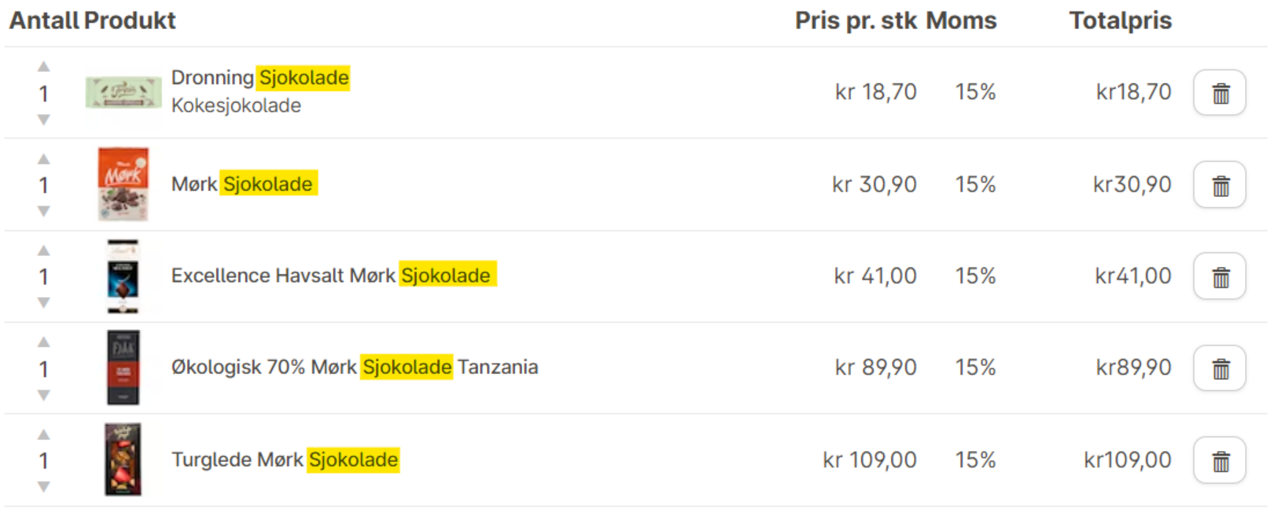
The results obtained from Müller [2021] are of interest in this thesis, as the data used in his study are the same as I will use when exploring classification. He investigated the potential for SSB to automate parts of the classification process of consumer goods into 5-digit COICOP codes for the consumer expenditure survey. He analyzed where misclassifications occurred and attempted to identify the underlying reasons, proposing a human-in-the-loop solution. The study used item descriptions and 5-digit COICOP 2018 codes as variables, which allowed the model to learn correlations between the two. This limited the model's ability to predict 5-digit COICOP codes based on variables other than the item descriptions. Müller [2021] discussed the limitations of applying an automatic classification system for scanned receipts, attributing errors and misclassifications to a lack of representative training data. Some 5-digit COICOP codes lacked sufficient training data, while others had limited data due to joined datasets with different classifications for identical text descriptions. Müller [2021] concluded that the absence of representative training data was the primary limiting factor in implementing a reliable automatic classification system.

1.3 My Approach

Some of the limitations mentioned by Müller [2021] are connected to the corpus of item descriptions and lack of training data. He approaches classification with discriminative eyes, making classification of items difficult as the corpus makes for ambiguous decisions. This is because a labeled dataset provided by SSB has multiple wrongly classified items and because of the natural structure of item descriptions in receipts. It will never be a perfect classification when classifying using machine learning techniques as in Benedikt et al. [2020] and Müller [2021], as there will always be a degree of uncertainty in these algorithms as the corpus of training data is at fault.

With data as the underlying problem, we can change our view on how to solve this classification task. By looking at classification from a generative point of view, I approach item classification based on the item descriptions. I suggest that corpus- and feature engineering is the key solution to improve the performance of classification tasks and that this is handled naturally by the generative approach.

To build on this concept, I use the idea of entity resolution and maximum entropy classification as a formal framework and suggest a solution involving an entity forest. By building trees from item descriptions, where we select root terms and branches to build the total item description, we can catch nuances between groups and items. Then when classifying an item from its item description, we can map the terms in the description to existing trees and classify a new item based on a score from tree-matching. I illustrate this idea by looking at different chocolate products available from an online Norwegian grocery retailer shown in Figure 3.



Antall	Produkt	Pris pr. stk	Moms	Totalpris
1	Dronning Sjokolade Kokesjokolade	kr 18,70	15%	kr18,70
1	Mørk Sjokolade	kr 30,90	15%	kr30,90
1	Excellence Havsalt Mørk Sjokolade	kr 41,00	15%	kr41,00
1	Økologisk 70% Mørk Sjokolade Tanzania	kr 89,90	15%	kr89,90
1	Turglede Mørk Sjokolade	kr 109,00	15%	kr109,00

Figure 3: Grocery list from the online grocery retailer, Oda.com/no.

The chocolate products all have the term ’sjokolade’ in the item descriptions and all these items belong to the 5-digit COICOP code ’01.1.8.5’ - *Chocolate, cocoa and cocoa-based food products* as shown in Figure 4.

The five different item descriptions in the grocery list all share the term ’sjokolade’, but three of them also share the word ’mørk’. I list up the different item descriptions below

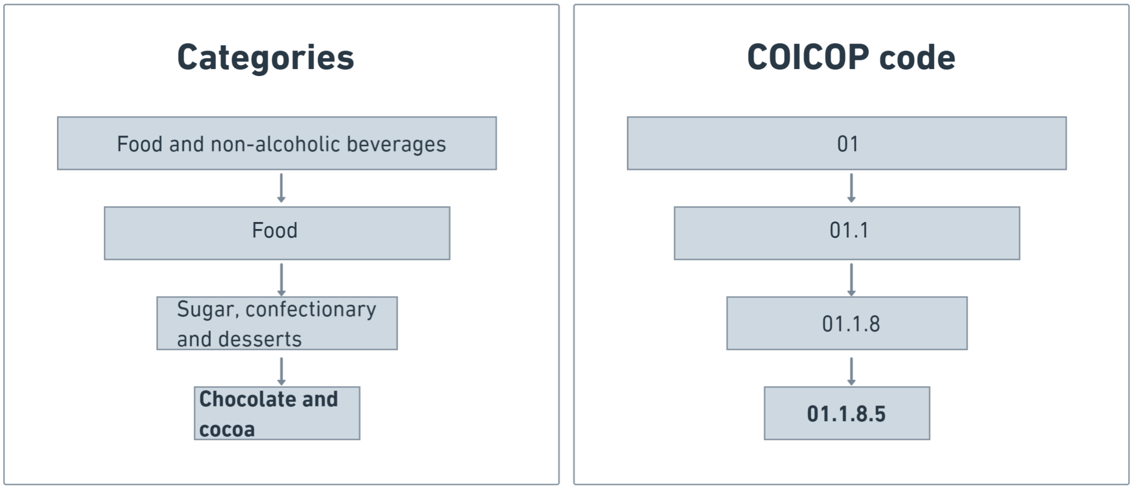

Categories
Food and non-alcoholic beverages
Food
Sugar, confectionary and desserts
Chocolate and cocoa
COICOP code
01
01.1
01.1.8
01.1.8.5

Figure 4: Classification of chocolate.

and build a tree from the text descriptions shown in Figure 5.

1. {excellence havsalt mørk sjokolade }

2. {økologisk mørk sjokolade tanzania}

3. {mørk sjokolade }

4. {dronning sjokolade}

5. {turglede mørk sjokolade}

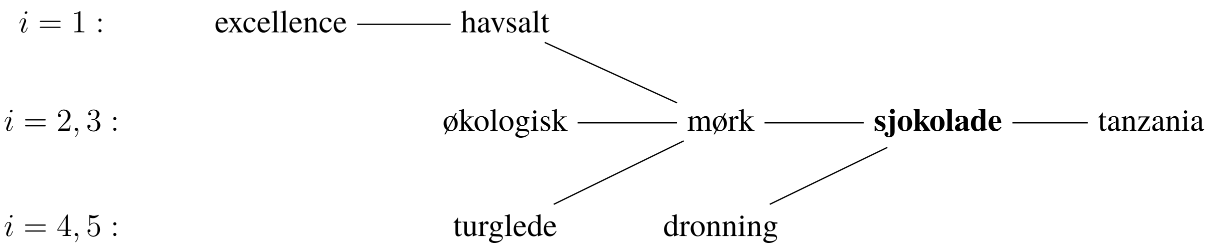


Figure 5: Tree for 5-digit COICOP code ’$01.1.8.5$’

The tree built consists of one root term, ’sjokolade’, which all of the item descriptions share. The other terms are labeled as branches of the tree. Now classification is a case of mapping terms of an item description to root and branches in the entity forest. For an item { dronning sjokolade} we will have a root-match on the term ’sjokolade’ and branch-match on the term ’dronning’ and a match score will be given accordingly.

1.4 Aims of Study

The previous attempts by SSB to develop an automated classification method to reduce manual work have been misguided. Rather than solely focusing on the classification techniques, more attention should have been given to the dataset itself, as the quality of the data and previous data-related work can pose challenges for classification using AI techniques. Engaging in feature engineering and transformation of the corpus is therefore seen as crucial to enhance classification accuracy. In this thesis, I aim to engineer the corpus to construct the entity forest, using entity resolution to improve the classification. By using a generative approach with corpus engineering, achieving true resolution and more accurate classification should be feasible. For the case of uncertainty in regard to classification, I propose a human-in-the-loop solution.

2 Theoretical Framework

The data cleaning, engineering, building, and classification performed in this thesis were conducted using multiple software tools. The tools used were Python version 3.9.7, RStudio version 2022.07.1 and ChatGPT version 3.

RStudio was used for cleaning and processing of data performed in Section 3.2. Python was used for building and classification of the entity-forest done in Section 3.4 and Section 4. Lastly, ChatGPT version 3 has been used as a tool to understand and solve warnings- and error messages regarding code for preprocessing, building, and classification.

This section provides an in-depth exploration of the data used in COICOP classification and the various methods and ideas implemented to build upon the concept of the entity forest; entity resolution, generative thinking, and maximum entropy.

2.1 Data

The data used in this thesis is provided by SSB, which is sourced from multiple origins and was transformed to a usable format in the study by Müller [2021]. The following section is a description of the origins of the data sources and the transformations which were done. This section is not a direct quote but closely linked to the description in his thesis *Classification of Consumer Goods into 5-digit COICOP 2018 Codes* [Müller, 2021].

Contrary to what Müller [2021] did, I will only look at parts of these datasets as a proof-of-concept to show the advantages of feature engineering. Any comparisons I make to Müller [2021]'s works will be done on the same extracted data using the algorithms from his study provided by SSB.

2.1.1 Data Origins

In the study done by Müller [2021], he used a variety of datasets that contained consumer goods and different item category codes, where the only requirement was that they were to have labeled items in text format. This led to a selection of data sets with different characteristics, but where all share this feature. The following five datasets were used:

- **Receipts:** Entries of consumer goods extracted from images of receipts or manually registered in SSB's phone app. SSB's phone app is used by households to either scan their receipts or manually write in their expenses.
- **Keywords:** Consists of 5-digit COICOP codes and a set of common consumer goods related to each code.
- **Transactions:** Entries of consumer goods registered as purchases by Norwegian grocery stores.
- **Imports:** Includes entries of consumer goods registered as imports by Norwegian customs.

The Receipts dataset was collected by SSB during their pilot study for the Consumer Expenditure Survey 2022. It contains entries of purchased consumer goods with some corresponding 5-digit COICOP codes. The items with valid codes were manually registered into SSB's phone app, while items without codes were extracted from scanned receipts. The dataset contains various features such as store names, price of items, and date of purchase.

The Keywords dataset is the Norwegian translation of the list of COICOP-2018 groups and corresponding examples provided in the official COICOP-2018 manual. It was used for auto-completion of expense registrations in SSB's phone app. Respondents register expenses, which are compared with similar item names in the Keywords dataset. The registration is automatically labeled with the corresponding 5-digit COICOP code if a match is found. This dataset includes consumer goods descriptions, the corresponding 5-digit COICOP codes, and item names.

The Transactions dataset combines multiple datasets prepared and merged by SSB to create a COICOP dataset with 5-digit COICOP codes. The main components of the dataset are transaction data from Norwegian grocery stores and product catalogs used in previous calculations of Consumer Price Indices (CPI). The transaction data was from 2018, while the product catalog data was updated for 2021.

The CPI dataset includes labeled non-food items from SSB. These items are additional consumer goods that were not previously included in the COICOP 2018 dataset. The dataset contains items labeled using the ECOICOP coding structure, which refers to the European COICOP 2016 coding system, a 6-digit code used by the CPI department at SSB to categorize items. Most of the data is from the CPI group, where entries have been partially labeled manually and partly generated from machine learning predictions with high prediction probabilities.

The Imports dataset is created from individual customs declarations registered with TVINN, the Norwegian customs' electronic system for exchanging customs declarations. This dataset includes imported goods from 2018 and their corresponding code in the CN 2008 coding format. The CN code is an 8-digit coding framework used for classifying goods for common custom tariffs. Unlike COICOP, the CN code is also used to categorize items that are not necessarily intended for consumption.

2.1.2 Preparations and Combining Dataset

Below I chose to summarise the steps done by Müller [2021] to prepare the dataset for use

- **Preparations**
  In this stage, each dataset was processed to extract a subset containing the entries of consumer goods and the corresponding 5-digit COICOP codes. A subset was made for the *Receipts* and *Keywords* datasets by extracting the columns for item names and the COICOP codes. The rows where the COICOP code was either missing or invalid were removed. The result was two prepared subsets of 568 and 2,377 entries.

  The *Transactions* dataset was already used by SSB and was already cleared for use in COICOP classification. The *CPI* dataset had multiple unlabelled entries, which were removed. The CPI dataset has used the ECOICOP coding system, which is different from the 5-digit COICOP coding. SSB has a conversion table to map ECOICOP codes to COICOP codes, using either a many-to-one or a one-to-one correspondence between the two formats. This conversion table translated the codes into 5-digit COICOP 2018 codes. The columns which did not contain item names and COICOP codes were removed from both datasets, leaving two subsets of 29,776 and 23,541 entries. The Imports dataset was processed to remove duplicate item name entries, and the relevant items were transformed into the 2018 format 5-digit COICOP codes using Eurostat's conversion tables.

The *Imports* dataset contained 18,030,591 item entries, most of these were duplicates, and deleting duplicates reduced the dataset to 3,378,407 entries. This dataset is different from the other datasets due to its CN 2008 coding format. Entries in the Imports dataset also included items that were intended for production, not only consumption. COICOP classification is only about items that are intended for consumption, therefore items used in production were not relevant. Preparation of the Imports data set involved a transformation of the coding format of the relevant items into the 2018 format 5-digit COICOP codes. Available conversion tables were not used due to loss in transformation. Müller [2021] developed a custom search algorithm to identify possible matches at lower detail levels to prevent zero matches between the transformation steps. With the custom search algorithm, a larger part of the Imports data set was transformed. This resulted in a prepared Imports subset of 1,433,947 entries with item names and 5-digit COICOP codes.

- **Combining**

  Once the individual data subsets were prepared, they were combined into a dataset by vertically concatenating them. This resulted in a combined dataset consisting of 1,490,216 rows and two columns.

2.2 Entity Resolution

Entity resolution is in this text is used as a collective term for similar concepts such as entity matching, data matching, record matching, and record linkage. All refer to the idea of finding records matching the same real-world entity. Entity resolution is useful when joining different datasets based on entities that do not share the same identifier. Different identifiers could result from discrepancies in record shapes, databases, storage location, or misspelled entities [Lee, Zhang, and Kim, 2021]. For example, name and date of birth may not exclusively identify a person as a result of two individuals with the same name having the exact date of birth. However, adding a current address will allow for unique identification. This way, entity resolution can help companies make inferences across large volumes of data in their databases or against other databases to bring together records corresponding to the same entity [Winkler, 2014]. Large parts of entity resolution consist of combining, preprocessing, cleaning, and feature engineering before matching occurs.

The idea of linking records in official statistics has been implemented and used for several decades. In 1985 Jaro [1989] used record-linkage methodology when matching individuals counted in a census to those counted in an independent post-enumeration survey. In 2015, Abbott, Jones, and Ralphs [2015] made approaches to linking records to support censuses using traditional methods and linking administrative data across population registers. The classical approach was developed by Fellegi and Sunter [1969]; they presented a probabilistic decision rule for record linkage, which was a method used to identify and merge records that referred to the same entity across different datasets. This is by proposing a likelihood ratio test approach to determine the probability that a given record pair was a true match.

We apply the idea of entity resolution to Household expenditure data where we want to join item descriptions of consumer goods to the 5- digit COICOP codes. The COICOP codes are considered entities and the item descriptions are the records. When we want to assign a record to an entity, we have the advantage of knowing the entities from the labeled data, unlike the most common entity resolution situation where the true underlying entities are unknown.

2.2.1 Classification as Entity Resolution

In the following sections, the term "groups" will be used to refer to the 5-digit COICOP codes. For classification based on entity resolution, we propose the following notations and descriptions of our data:

- $i = 1, ..., N$ are the *items* to be classified. Let $U = \{1, ..., N\}$.
- $y = 1, ..., K$ as the *groups* to which items are classified. Let $\Gamma = \{1, ..., K\}$.
- x is any *term* that can be used in item description.
- $\boldsymbol{x}$ as the collection of terms in *item description*
- Ω as the *corpus* of item description, i.e.

$$\Omega = \{\boldsymbol{x}_i : i \in U\}$$

For an item in our corpus: *freia mørk sjokolade*:

Notation	**Item**
x	{freia}, {mørk}, {sjokolade }
$\boldsymbol{x}$	{freia, mørk, sjokolade }
y	01.1.8.5

For each $i \in U$, we let y_i be the group; 5-digit COICOP code. COICOP classification can then be viewed as an entity resolution problem, where Γ are the known entities and U the records. The records $U_y = \{i \in U : y_i = y\}$, are considered to be matched (to each other) via reference to the entity y. The *resolution* we seek is the partition $U = \bigcup_{y \in \Gamma} U_y$, denoted by

$$\mathbb{C} = \{U_y : y \in \Gamma\}$$

2.3 Generative Approach

This section describes our approach to entity resolution, which is guided by a generative mindset. Within supervised machine learning, there are different approaches to learning, we refer to them as *generative* and *discriminative* [Ng and Jordan, 2001].In the generative approach to classification, we aim to model the joint distribution, $p(x, y)$, of the input x and the groups y from training data. Then we compute the conditional densities for each group and assign the input x to the group y with the highest posterior probability. With a discriminative approach, classification can directly model the posterior density $p(y|x)$, which is then used to classify a new instance.

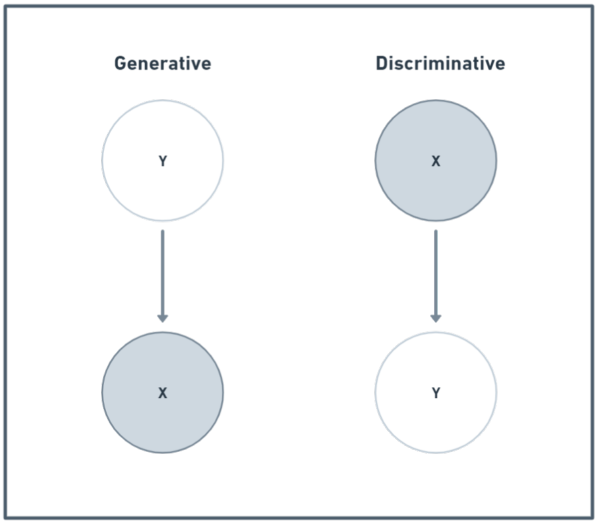


Figure 6: Generative vs. discriminative classification.

In the graph displayed in Figure 6, we aim to show how the structure of a generative and discriminative model works. The direction of the lines denotes what probabilities we can infer [Causevic, 2021]. Our data has an input $\boldsymbol{x}$, being the item descriptions and y as groups. Generative models focus on estimating the probability distribution of the input data conditioned on group, while discriminative models aim to map inputs to groups directly.

In the previous work done by Müller [2021] for SSB, he explores the discriminative classifiers in machine learning; logistic regression and random forest. By the discriminative approach, classification of y_i for any item $i \in U$ is based on

$$f(y|\boldsymbol{x};\Omega) = \Pr(y_i = y \mid \boldsymbol{x}_i = \boldsymbol{x};\Omega)$$

where the different terms in the item description $\boldsymbol{x}$, are used as distinct features for $f(y|\boldsymbol{x};\Omega)$. The model function, given as $f_U(y|\boldsymbol{x};\Omega)$, is determined based on the corpus Ω and the true resolution $U_y : y \in \Gamma$. However, if any term, such as $x =$ 'økologisk', appears in multiple item descriptions belonging to different 5-digit COICOP codes, classification of $\boldsymbol{x}_i$ containing this x-term migth be wrong by the *discriminative classifier*

$$y_i = \arg\max_{y \in \Gamma} \; f_U(y|\boldsymbol{x}_i;\Omega)$$

For the data used, we can see that the case of a term x appearing in multiple different 5-digit COICOP codes is normal. For example, in '$01.1.8.5$' - *Chocolate, cocoa, and cocoa-based food products*, '$01.1.8.6$' - *Ice, ice cream and sorbet* and '$01.1.8.9$' - *Other sugar confectionery and desserts*, 'sjokolade' and the English version 'chocolate' are frequently used amongst all groups. To solve this problem, we introduce the *generative approach*, where we focus on building the model function

$$f(\boldsymbol{x}|y;\Omega) = \Pr(\boldsymbol{x}_i = \boldsymbol{x} \mid y_i = y;\Omega)$$

Let $f_U(\boldsymbol{x}|y;\Omega)$ be the model function given the corpus Ω, and the true resolution $\{U_y : y \in \Gamma\}$. The corpus, Ω , should be *free of entity-duplication*, so that for any $\boldsymbol{x} \in \Omega$, we have

$$\sum_{y \in \Gamma} \mathbb{I}\big(f(\boldsymbol{x}|y;\Omega) > 0\big) \equiv 1 \tag{1}$$

So that two (or more) identical item descriptions only exist within one 5-digit COICOP code. This *admissibility* condition (1) is necessary for any well-defined mapping from the item descriptions Ω to the groups Γ. If we allow for mapping to multiple groups, classification becomes impossible. Note that we allow multiple items with the same $\boldsymbol{x}$ as long as they belong to the same group. Given any corpus Ω satisfying (1), classification of any $i \in U$ based on $\boldsymbol{x}_i$ would always be correct by the *generative classifier*

$$y_i = \arg\max_{y \in \Gamma}\ f_U(\boldsymbol{x}_i | y; \Omega)$$

regardless if terms exist in multiple item descriptions belonging to different groups.

2.4 Maximum Entropy

2.4.1 Maximum Entropy Principle

Maximum entropy is a method for determining probability distributions from data by assuming that when there is no information, the distribution is as uniform as possible (maximum entropy). It uses labeled training data to establish constraints for the model, represented as expected values of "features". In simple terms, the intuitive principle behind maximum entropy is to model what is known without making assumptions about what is unknown. So, when given a set of facts, the goal is to choose a model that is consistent with all the facts while being as uniform or unbiased as possible in other aspects [Berger, Della Pietra, and Della Pietra, 1996]. When maximizing entropy without knowledge of our data, the universal constraint states that the sum of probabilities equals 1. Then the maximum entropy probability distribution is the uniform distribution, where each event has an equal probability of occurrence,

$$p_i = \frac{1}{n}, \quad \forall i \in 1, \ldots, n.$$

In an event where we are given constraints, for example, we have four food groups, and we are told that about 40 % of item descriptions containing the word "milkshake" belongs to the 'milk-based desserts and beverages'- group. When given a text with 'milkshake', we would assume that there is a 40 % chance of it being in the 'milk-based desserts and beverages' group and a 20% chance of being in one of the other food groups. If a text does not contain 'milkshake', we would assume a uniform distribution with a 25 % chance of each group. This is an example of how the maximum entropy model works with given constraints; the model can take on more constraints and can be used to estimate all probability distributions [Nigam, Lafferty, and McCallum, 1999].

2.4.2 Maximum Entropy Classification

As proposed done in Lee et al. [2021], we adapt the maximum entropy method for classification to entity resolution. Specifically, looking at the likelihood ratio method proposed by

Fellegi and Sunter [1969] as a special case of the ratio and applying the maximum entropy method for ratio estimation. For *maximum entropy classification (MEC)*, we propose to use the probability ratio

$$r_y(\boldsymbol{x}_i) = \frac{\Pr(\boldsymbol{x}_i \mid i \in U_y)}{\Pr(\boldsymbol{x}_i \mid i \in U)} := \frac{f(\boldsymbol{x}_i|y_i;\Omega)}{f(\boldsymbol{x}_i;\Omega)}$$

where the denominator is the marginal probability over all the items U, and the numerator is the conditional probability given $i \in U_y$, and then maximise the Kullback-Leibler divergence (relative entropy) from $f(\boldsymbol{x})$ to $f(\boldsymbol{x}|y)$;

$$D_{KL}(\mathbb{C};\Omega) = \sum_{y\in\Gamma}\sum_{i\in U_y} f(\boldsymbol{x}_i|y;\Omega)\log r_y(\boldsymbol{x}_i) \tag{2}$$

subjected to the constraint that $f(\boldsymbol{x}|y;\Omega)$ defines a conditional distribution, i.e.

$$\sum_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}|y;\Omega) = 1, \quad \forall y \in \Gamma.$$

From our labelled dataset, both $f(\boldsymbol{x};\Omega)$ and $f(\boldsymbol{x}|y;\Omega)$ are given. So to classify an item i with item description $\boldsymbol{x}_i = \boldsymbol{x}$ we look at the difference of D when assigning $\boldsymbol{x}$ to two different groups; $y_i = y$ or $y_i = y'$.

$$\begin{aligned}
&f(\boldsymbol{x}|y)\log\big\{f(\boldsymbol{x}|y)/f(\boldsymbol{x})\big\} - f(\boldsymbol{x}|y')\log\big\{f(\boldsymbol{x}|y')/f(\boldsymbol{x})\big\}\\
&= f(\boldsymbol{x}|y)\log f(\boldsymbol{x}|y) - f(\boldsymbol{x}|y')\log f(\boldsymbol{x}|y') + \big\{f(\boldsymbol{x}|y) - f(\boldsymbol{x}|y')\big\}\log f(\boldsymbol{x})\\
&= f(\boldsymbol{x}|y)\big\{\log f(\boldsymbol{x}|y) - \log f(\boldsymbol{x}|y')\big\} + \big\{f(\boldsymbol{x}|y) - f(\boldsymbol{x}|y')\big\}\{\log f(\boldsymbol{x}|y') - \log f(\boldsymbol{x})\}
\end{aligned}$$

It follows that, provided

$$\min\{f(\boldsymbol{x}|y),\ f(\boldsymbol{x}|y')\} \geq f(\boldsymbol{x})$$

the Kullback-Leibler divergence (2) is higher given $y_i = y$ than $y_i = y'$ if and only if $f(\boldsymbol{x}|y) > f(\boldsymbol{x}|y')$, where a higher Kullback-Leibler divergence indicates a larger divergence or dissimilarity between the two probabilities being compared. In other words, MEC yields generally the generative classifier

$$y_i = \arg \max_{y \in \Gamma : f(\boldsymbol{x}_i|y) \geq f(\boldsymbol{x}_i)} f(\boldsymbol{x}_i|y;\Omega)$$

which holds as well for the special case where the corpus Ω satisfies the condition (1).

2.4.3 Corpus Purity and Transformation

Feature extraction is a widely used technique in natural language processing that aims to understand the meaning of texts and identify important characteristics. In our approach, we utilize the concept, which we name *text purity*, to understand the corpus and its important terms.

We propose that for any $i \neq j \in U_y$ where $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ for a given corpus Ω satisfying (1) where all item descriptions are unchanged and distinct, and for a modified corpus Ω' we let item descriptions be identical; $\boldsymbol{x}_i = \boldsymbol{x}_j = \boldsymbol{x}$. This modified corpus Ω' have a higher purity, as there are fewer distinct item descriptions in Ω' than Ω. When this is the case, we can also observe that the KL-divergence (2) is increased when comparing Ω' to Ω, i.e., $D_{KL}(\mathbb{C};\Omega') > D_{KL}(\mathbb{C};\Omega)$.

$D_{KL}(\mathbb{C};\Omega)$ calculated for the two items i and j

$$\frac{Nf(\boldsymbol{x}_i)}{N_y}\log\frac{Nf(\boldsymbol{x}_i)/N_y}{f(\boldsymbol{x}_i)} + \frac{Nf(\boldsymbol{x}_j)}{N_y}\log\frac{Nf(\boldsymbol{x}_j)/N_y}{f(\boldsymbol{x}_j)} = \big(f(\boldsymbol{x}_i) + f(\boldsymbol{x}_j)\big)\frac{N}{N_y}\log\frac{N}{N_y}$$

where $N_y = |U_y|$

$D_{KL}(\mathbb{C};\Omega')$ calculated for the two items i and j

$$2\big(f(\boldsymbol{x}_i) + f(\boldsymbol{x}_j)\big)\frac{N}{N_y}\log\frac{N\big(f(\boldsymbol{x}_i) + f(\boldsymbol{x}_j)\big)/N_y}{f(\boldsymbol{x}_i) + f(\boldsymbol{x}_j)} = 2\big(f(\boldsymbol{x}_i) + f(\boldsymbol{x}_j)\big)\frac{N}{N_y}\log\frac{N}{N_y}$$

The corpus with the highest purity is given by Lemma 1.

Lemma 1. *The corpus* Ω *of the highest purity, i.e.* $f_U(\boldsymbol{x}_i|y;\Omega) \equiv 1$ *for any* $i \in U_y$ *and* $y \in \Gamma$, *maximises the KL-divergence* (2) *among all the corpus's satisfying* (1).

In the example below, we show three different versions of a corpus; Ω_1,Ω_2 and Ω_3, and look at the difference in D_{KL}

$\boldsymbol{x} \in \Omega_1$	$\boldsymbol{x} \in \Omega_2$	$\boldsymbol{x} \in \Omega_3$	y
dronning sjokolade	sjokolade	sjokolade	01.1.8.5
økologisk mørk sjokolade	sjokolade	sjokolade	01.1.8.5
lindor assortert sjokolade	sjokolade	sjokolade	01.1.8.5
laks frossen	laks frossen	laks	01.1.3.1
fersk laks	fersk laks	laks	01.1.3.1
salma laks koteletter	salma laks koteletter	laks	01.1.3.1
krone is jordbær	krone is jordbær	is	01.1.8.6
karamell iskrem	karamell iskrem	iskrem	01.1.8.6
bringebær sorbet	bringebær sorbet	sorbet	01.1.8.6
lollipop	lollipop	lollipop	01.1.8.6

$$\text{For } \Omega_1: D_{KL} = 3\cdot\left(\tfrac{1}{3}\right)\cdot\log\left(\frac{\left(\frac{1}{3}\right)}{\left(\frac{1}{10}\right)}\right)+3\cdot\left(\tfrac{1}{3}\right)\cdot\log\left(\frac{\left(\frac{1}{3}\right)}{\left(\frac{1}{10}\right)}\right)+4\cdot\left(\tfrac{1}{4}\right)\cdot\log\left(\frac{\left(\frac{1}{4}\right)}{\left(\frac{1}{10}\right)}\right) = 3.324236$$

$$\text{For } \Omega_2: D_{KL} = 3\cdot\left(\tfrac{3}{3}\right)\cdot\log\left(\frac{\left(\frac{3}{3}\right)}{\left(\frac{3}{10}\right)}\right)+3\cdot\left(\tfrac{1}{3}\right)\cdot\log\left(\frac{\left(\frac{1}{3}\right)}{\left(\frac{1}{10}\right)}\right)+4\cdot\left(\tfrac{1}{4}\right)\cdot\log\left(\frac{\left(\frac{1}{4}\right)}{\left(\frac{1}{10}\right)}\right) = 5.732182$$

$$\text{For } \Omega_3: D_{KL} = 3\cdot\left(\tfrac{3}{3}\right)\cdot\log\left(\frac{\left(\frac{3}{3}\right)}{\left(\frac{3}{10}\right)}\right)+3\cdot\left(\tfrac{3}{3}\right)\cdot\log\left(\frac{\left(\frac{3}{3}\right)}{\left(\frac{3}{10}\right)}\right)+4\cdot\left(\tfrac{1}{4}\right)\cdot\log\left(\frac{\left(\frac{1}{4}\right)}{\left(\frac{1}{10}\right)}\right) = 8.140128$$

By maximizing $D_{KL}(\mathbb{C};\Omega)$ over Ω one can use feature engineering to engineer a corpus Ω^* with a higher entity purity and build a function $t : \Omega \rightarrow \Omega^*$, that maps each item description in the original corpus Ω to one in the engineered corpus Ω^*, denoted by $\boldsymbol{x}^* = t(\boldsymbol{x}) \in \Omega^*$ given $\boldsymbol{x} \in \Omega$.

We use the example given above to prove our case. Suppose the entity purity of corpus is increased from a sample of Ω to Ω^* for two items in the 5-digit code '01.2.1.0'

- $\boldsymbol{x}_i = \{\text{økologisk, mørk, sjokolade}\} \quad \rightarrow \quad \boldsymbol{x}_i^* = \{\text{sjokolade}\}$
- $\boldsymbol{x}_j = \{\text{dronning, sjokolade}\} \quad \rightarrow \quad \boldsymbol{x}_j^* = \{\text{sjokolade}\}$

This makes sense intuitively because both the items are chocolates (sjokolade), while the other terms such as 'økologisk' or 'dronning' are not the defining characteristics for the group classification of interest. Similarly, suppose the following in a different 5-digit code '01.1.5.3':

- $\boldsymbol{x}_k = \{\text{lollipop}\} \quad \rightarrow \quad \boldsymbol{x}_k^* = \{\text{lollipop}\}$
- $\boldsymbol{x}_l = \{\text{ karamell, iskrem}\} \quad \rightarrow \quad \boldsymbol{x}_l^* = \{\text{iskrem}\}$

Then conditional probabilities yield

$$f(\text{sjokolade}|01.2.1.0; \Omega^*) = 1$$

and

$$f(\text{lollipop}|01.1.5.3; \Omega^*) = f(\text{iskrem}|01.1.5.3; \Omega^*) = 1/2.$$

Suppose now for Ω^*, that classification is needed for a new item description introduced: $\boldsymbol{x} = \{\text{sjokolade, iskrem}\}$. The classification of this item description would be correct if feature engineering of Ω^* would give $\boldsymbol{x} = \{\text{iskrem}\}$ but not if $\boldsymbol{x} = \{\text{sjokolade}\}$. This is of essence when we now move on to our next section where we propose a solution for the transformation from Ω to Ω^*.

3 Entity Forest for Entity Resolution

Based on our previous deductions, we aim to create a model for transforming our corpus, denoted $t : \Omega \to \Omega^*$. Our proposed solution involves constructing an entity forest of item descriptions. This method allows us to effectively resolve the entities within the corpus, turning tree-building into a form of feature engineering. Classifying item descriptions into trees now becomes a matter of entity resolution, with potential challenges arising from data sparsity.

In this section, I give the assumptions necessary for building and classifying with the entity forest model. Next, I dive into the building process using the dataset provided by SSB, which was processed as explained in Section 2.1.

3.1 Assumptions for Entity Forest

I propose an *entity forest* model for $t : \Omega \to \vec{\Omega}$ as the means to entity resolution. Where $\vec{\Omega}$ is the transformed corpus.

1. Both Ω and $\vec{\Omega}$ should be *free of entity-duplication*. The terms in an item description are treated as sequential in $\vec{\Omega}$, i.e., ordering the terms in an item description matters.

2. The transformation from each $\boldsymbol{x} \in \Omega$ to $t(\boldsymbol{x}) \in \vec{\Omega}$ is determined by a tree (of terms) in the entity forest, which needs to be appropriately chosen for the given $\boldsymbol{x}$.

3. Each tree in the entity forest pertains to one and only one entity in Γ; but there are usually multiple trees for each given entity. Choosing the tree for $\boldsymbol{x}$ is the same as classifying y given $\boldsymbol{x}$.

4. Each tree has a *root* term. All the root terms of $\vec{\Omega}$, denoted by $\text{root}(\vec{\Omega})$, is a corpus free of entity-duplication and $\vec{\Omega}$ has a higher *root purity* than Ω in that $D\big(\mathbb{C}; \text{root}(\vec{\Omega})\big) > D(\mathbb{C}; \Omega)$ by $D(\mathbb{C}; \Omega)$.

3.2 Preprocessing and Data Cleaning

For this thesis I have chosen to look at a section of the data, for the 2-digit COICOP code ’01’ - *food-and non-alcoholic beverages*. From the total dataset of 1,048,575 observations, item descriptions within ’01’ stands for 11% of the data, with 16 different 4-digit COICOP codes within, as shown in Figure 7.

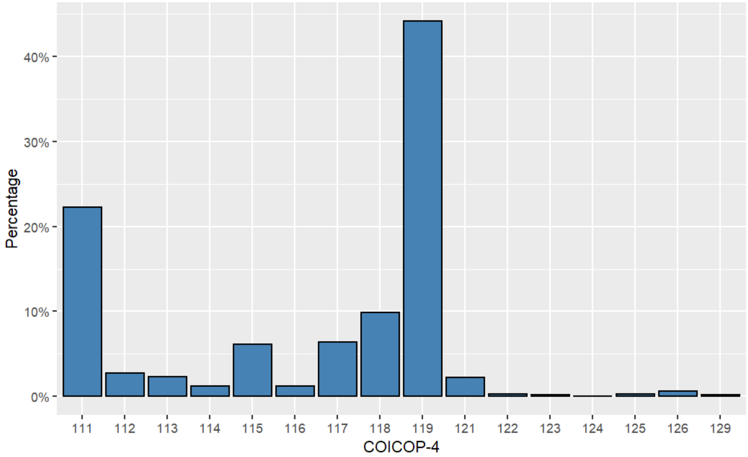


Figure 7: 4-digit COICOP codes within ’01’.

Before building trees from item descriptions we modify the dataset as to easier access the important terms.

- **Numbers:** Remove all numbers.
- **Lowercase:** Change all text descriptions to lowercase.
- **Special Characters:** Remove all occurrences of special characters of the form: [/ . , ; ‘ > < + - * ? \]
- **Newline Characters:** Remove newline character ("\n")
- **Single Characters:** Remove characters that occur alone.
- **Single Characters:** Remove stop-words.

Some stop words were excluded; stop-words refer to frequently used words in a language, such as "a", "the", "is", and "are". These are often removed during text mining and natural

language processing (NLP) due to their limited informative value [Ganesan, 2019]. In the Norwegian language, this includes words as 'for', 'med', 'uten', 'og'. In the import data, we observe that many of the item descriptions include 'ank', 'dato', 'parcel', 'container-nummer', we remove these as well as the following stop-words which are frequent within certain 4-digit COICOP codes:
'kk','pos', 'bx', 'ex' 'vv', 'av', 'alu', 'for', 'pk', 'plu', 'pe', 'vac', 'mk', 'kll', 'co', 'cs'.

3.3 Entity- duplication

To ensure that our resolution is correct, it is essential that identical items in our dataset are classified to the same 5-digit COICOP code. The easiest approach is to make certain that all identical item descriptions appear within the same 5-digit code, so that Ω and $\vec{\Omega}$, are *free of entity-duplication*. For an item description x_i, every item identical to x_i should appear within the same 5-digit code, so that all instances of x_i are distinct. If these identical items are classified into different codes, classifying instances to trees is based on an ambiguous dataset, and classification can not be correct. For this condition, we know that for our labeled dataset, in all instances where identical item descriptions have the same 5-digit COICOP code, the resolution for our dataset can be correct. We take a look at the dataset provided by SSB, this dataset is partially manually classified, and the rest is classified using different thresholds. This leads to our first problem; manual labeling can lead to wrongly classified items by human error. From our raw input data, this seems to be the case. We have several identical item descriptions with different 5-digit COICOP codes.

From Table 1, 'agurk' is classified in five instances. In four of them to 5-digit COICOP code '01.1.7.2' - *Fruit-bearing vegetables, fresh or chilled*, and in one instance to 5-digit COICOP code '01.1.7.9' - *Vegetables, tubers, plantains, cooking bananas and pulses ground and other preparations*. The correct classification should be to 5-digit COICOP code '01.1.7.2' - *Fruit-bearing vegetables, fresh or chilled*, according to UN [2018]. For this case, we have multiple instances correctly classified and one wrongly classified. Below, I suggest three approached to solve for the problems of entity-duplications which appear in the given dataset.

	5-digit COICOP code	Item description	n
1	01.1.7.2	agurk	4
2	01.1.7.9	agurk	1
3	01.2.1.0	appelsinsaft	1
4	01.2.9.0	appelsinsaft	1
5	01.1.7.1	artisjokk	1
6	01.1.7.9	artisjokk	1
7	01.1.2.2	bayonneskinke	1
8	1.1.2.3	bayonneskinke	1
⋮	⋮	⋮	⋮
112	01.1.6.5	druer røde beger	82
113	01.1.7.4	druer røde beger	1
⋮	⋮	⋮	⋮
587	01.2.6.0	zeroh sitron lime	1
588	01.2.9.0	zeroh sitron lime	1

Table 1: Extract from identical item descriptions appearing in different 5-digit COICOP codes.

3.3.1 Approach for Solving Entity-duplications

1. **Norwegian 5-digit COICOP codes**

   We have a dataset provided by SSB that consists of all 5-digit COICOP codes given, with examples for each of the codes. This is a Norwegian translation of the UN [2018] given by the UN Statistics Division. An extract from this dataset is given in Table 2.

   We can correctly classify some items by comparing the duplicated item descriptions with the names given in the dataset with item examples from SSB. For example, 'artisjokk' is shown to belong in 5-digit COICOP code '01.1.7.1', so we change item descriptions of instance 'artisjokk' to correct 5-digit code code.

2. **Frequency**

   For each item, we reclassify all instances to the item description with the highest count of a 5-digit COICOP code. So for the case of 'druer røde beger', most belong to 5-digit COICOP code '01.1.6.5', as seen in Table 3

   For 'druer røde beger', we are left with the dataset as shown in Table 4.

5-digit COICOP code	Item description
01.1.1.1	hvetekorn
01.1.1.1	rugkorn
01.1.1.1	byggkorn
01.1.1.1	maiskorn
01.1.1.1	quinoa
01.1.1.1	bulgur
01.1.1.2	hvetemel
01.1.1.2	kakemiks
01.1.1.2	bakemiks
01.1.1.3	terter
01.1.1.3	pai
⋮	⋮
01.1.7.1	brokkoli
01.1.7.1	spinat
01.1.7.1	salat
01.1.7.1	artisjokk

Table 2: Extract from item examples in different 5-digit COICOP codes from SSB.

5-digit COICOP code	Item description	n
01.1.6.5	druer røde beger	82
01.1.7.4.	druer røde beger	1

Table 3: ’druer røde beger’ before reclassification.

This approach may not always result in the correct 5-digit COICOP code for an item.

3. **Deleting duplicates**

Our last option is deleting all the remaining instances of the duplicated items. This will affect the item descriptions where we have a similar amount belonging to each 5-digit COICOP code, this is the case for ’bayonneskinke’:

5-digit COICOP code	Item description	n
01.1.2.2	bayonneskinke	1
01.1.2.3	bayonneskinke	1

This means that the remaining instances of ’bayonneskinke’ we are left with in the dataset are the six items containing the word as seen in Figure 8

5-digit COICOP code	Item description	n
01.1.6.5	druer røde beger	83

Table 4: ’druer røde beger’ after reclassification.

5-digit COICOP code	Item description	n
01.1.2.2	bones *bayonneskinke* kokt	1
01.1.2.5	coop *bayonneskinke* kokt	1
01.1.2.2	kokt *bayonneskinke* fana	1
01.1.2.2	coop *bayonneskinke* ra	1
01.1.2.3	*bayonneskinke* hansse	1
01.1.2.2	*bayonneskinke* kokt bø	1

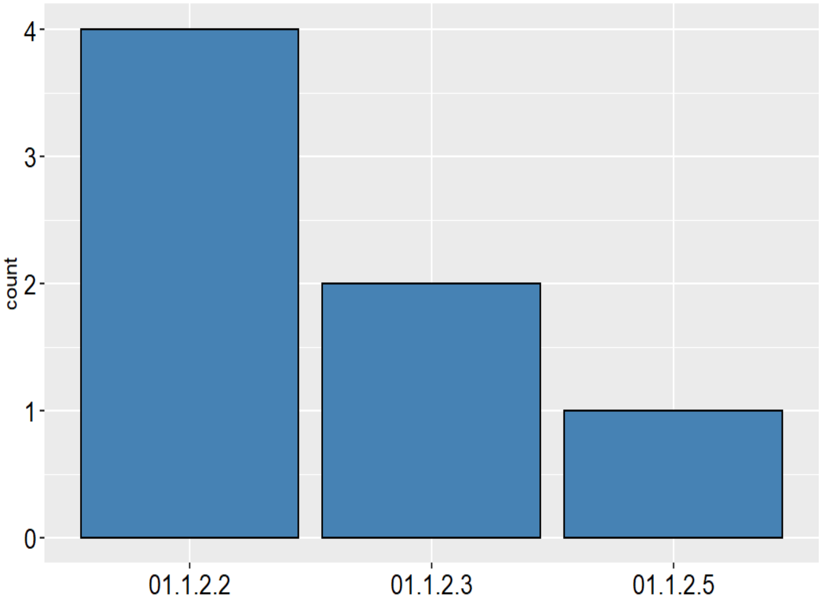


Figure 8: The three different 5-digit COICOP codes containing term ’bayonneskinke’.

5-digit COICOP code ’01.1.2.2’ - *Meat, fresh, chilled or frozen*, has the most instances of the item *bayonneskinke*. From noa [2022]:

> Bayonne ham or jambon de Bayonne is a **cured** ham that takes its name from the ancient port city of Bayonne in the far southwest of France.

From this, we can deduce that the correct 5-digit COICOP code should be ’01.1.2.3’ - *Meat, dried, salted, in brine or smoked.* As the dataset is wrongly classified from start due to the misclassification of labels on the original datasets from SSB, the classification can not be 100% correct. Solving for this problem might reduce misclassification later, but it will require human intervention. We will look at human-in-the-loop as a solution in Section 4, which can partly can be seen as a solution to this problem occurring.

3.3.2 Applying to our Dataset

We apply the three different approaches to our dataset in the same order as they are given. This, to remove duplicates in the dataset, where duplicates refer to two or more identical item descriptions classified to more than one 5-digit COICOP code.

We follow the steps for the dataset with 2-digit COICOP code ’01’: Food- and non-alcoholic beverages. For this dataset, we have a total of 1186 duplicated instances. For the steps given we reduce the number of duplications to zero. In Table 5, we have each step, the rows of duplications, and the number of duplications after each step.

Duplication steps	Rows of duplications	Duplications
start	1186	577
step 1	1041	510
step 2	604	298
step 3	0	0

Table 5: Duplication steps with number of duplications.

We delete 604 rows of item descriptions, 298 duplications, from 52 different 5-digit COICOP codes. The top five 4-digit COICOP codes where we delete instances are given in Figure 9. From the duplicated items that are deleted, the item descriptions where we delete most item descriptions, and so seemingly with the most difficult item descriptions to classify, belong to 4-digit COICOP code ’01.1.9’ - *Ready-made food.* Here the top three 5-digit COICOP codes are ’01.1.9.1’ - *ready-made food*, ’01.1.9.9’ - *other food products* and ’01.1.9.3’ - *salt, condiments and sauces.*

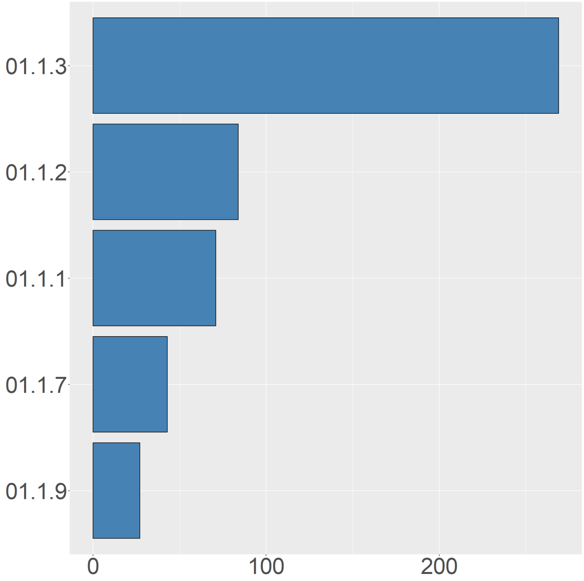


Figure 9: Top five 4-digit COICOP codes where we need to delete instances.

3.4 Building Entity Forest from Trees

In this section we will build our entity forest from our preprocessed dataset Ω by transforming each item description $\boldsymbol{x} \in \Omega$ to $t(\boldsymbol{x}) \in \vec{\Omega}$ so that we are left with $\vec{\Omega}$ as the transformed corpus and entity forest.

An important part of building an enitity-forest is understanding how the terms in the texts are structured. Sequence labeling of text and sentences is often a part of natural language processing and an important part of building a meaningful sentence. In the text descriptions, a sequential corpus $\vec{\Omega}$ is relevant regarding the classification of items, as some words in a text description bear a larger meaning than others. Text descriptions from grocery receipts do not necessarily stick to a template that follows a correct Norwegian grammatical structure, this can be confirmed by looking at product names in Norwegian stores or through grocery retailers online. Contrary to Norwegian sentence structure, adjectives/descriptives are placed both behind and in front of nouns. For example, for 'druer røde beger' in 5-digit COICOP code '$01.1.6.5$', if we are given a new instance $\boldsymbol{x}$: 'røde druer beger', 'druer' is a root term and we map $\boldsymbol{x}$ to a tree within 5-digit COICOP code '$01.1.6.5$', there will be a unique match on the term 'beger' after the root term. but we would not have a match on the text description as a whole. However, we know that these item description refers to the same item, this is a case of how we want to score mapping which we return to in Section 4.

Going forward with building trees for the entity forest, we revisit Assumption 2 of the entity forest, where we need to appropriately determine the tree of terms. For a given item description x, we build trees around the root term, as this term should hopefully be the term in the texts that contains the most meaning. For the case of 'røde druer beger', 'druer' is the bearer of the most meaninful term in the description and so referred to as a term in root($\vec{\Omega}$).

3.4.1 Building Trees from Roots

Choosing which rules to follow when building trees from root terms have been chosen by me. Other methods to building trees can be more appropriate, I mention some in Section 6.3. When building trees from root terms we follow the steps given below.

1. **Norwegian 5-digit COICOP codes**
   We use the Norwegian translation of the UN [2018] dataset provided by SSB that consists of all 5-digit COICOP codes, where we have example items for each code. When choosing these as root terms in the text descriptions, we can be sure that there exist no root duplications in root($\vec{\Omega}$). For example, in 5-digit COICOP code '01.1.2.3' - Meat, dried, salted, in brine or smoked, we have items shown in Table 6.

Item	Item name	5-digit COICOP code
1	{Bacon}	01.1.2.3
2	{Skinke}	01.1.2.3
3	{Fenalår}	01.1.2.3
4	{Spekeskinke}	01.1.2.3
5	{Salami}	01.1.2.3

Table 6: Item examples from different 5-digit COICOP codes

Which means that, for the 5-digit COICOP code '01.1.2.3', terms that coincide with these can be set as root words. Words before and after root words will be branches in the tree structure. Given four item descriptions:

- bøkerøkt skinke hel kokt
- serrano skinke skivet
- kokt skinke folkets
- kokt skinke jacobs

From these four item description we have the tree structure as illustrated in Figure 10. Colored lines are made to show the different item descriptions and are not a part of the true tree structure.

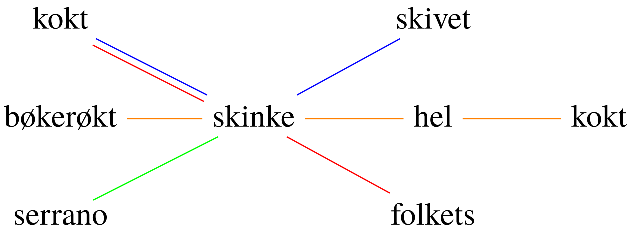


Figure 10: Example tree from item descriptions in '01.1.2.3'

2. **Frequency - term**

   Next we find the most frequent terms for each 5-digit COICOP code and use these as a base for the root terms. If not careful, it could lead to problems where frequent terms within one 5-digit code are likely to be frequent within other 5-digit codes, or that terms alone can not possibly be the root due to our assumptions of the root($\vec{\Omega}$) given in Assumption 4, that root($\vec{\Omega}$) is a corpus free of entity-duplication.

   For example, 'laks', is one of the most frequent terms within the 5-digit COICOP code '01.1.3.2' - Fish, dried, salted, in brine or smoked, with item descriptions such as 'gravet laks' and 'røkt laks'. The word 'laks' cannot be the root word, as it is the root term for another 5-digit COICOP code: '01.1.3.1 '- Fish, live, fresh, chilled or frozen. In Norwegian, 'gravet laks' is not naturally a compound and is difficult to catch if only looking at word frequency within 5-digit codes. Consequently, we introduce bi-gram frequency.

3. **Frequency - bi-grams**

   To better adjust for the situation where word frequency will break our assumption of root terms in $\vec{\Omega}$. This will happen when words are not naturally compounded and reappear in multiple 5-digit COICOP codes. Hence, we add the option that if a term chosen as root, already exists as a root from the previous steps, we look at the most frequent bi-grams within the 5-digit COICOP codes, and use both as terms as a root. See Table 7 and Figure 11 for the bigram solution for root term ’gravet laks’. 70% of the instances are given as ’gravet laks’ the last 30% as ’laks gravet’, which will be two different trees in the entity forest. For increasing root purity, we mention root rotation as a possible solution to this in Section 6.3.

Table 7: Example of an entity-fores from ’01.1.3.2’, with item descriptions from Ω

Item description	5-digit COICOP code	New ’compound’
{lofoten, gravet, laks, bit}	01.1.3.2	{lofoten, gravet laks, bit}
{gravet, laks, skiver}	01.1.3.2	{gravet laks, skiver}
{gravet, laks, bit, dolmoy, seafood}	01.1.3.2	{gravet laks, bit, dolmoy, seafood}

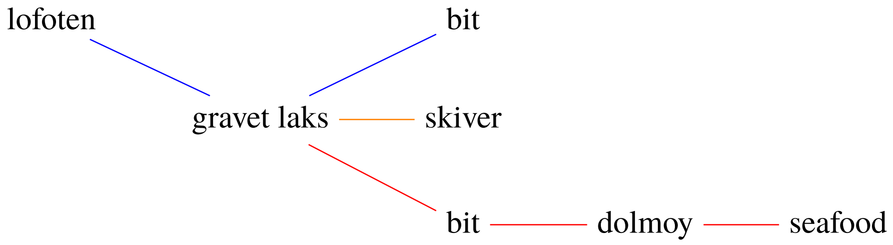


Figure 11: Example tree from three text descriptions in ’01.1.8.6’

We have now built the entity forest where each tree pertains to one and only one entity in Γ, with some 5-digit COICOP codes having more trees for each entity than others. In the next section we look at how a possible classification with an entity forest can be led.

4 Classification

Classification is carried out using our entity forest to look for agreement on the different terms by mapping terms in an item description to trees in the entity forest. To be able to classify instances to 5-digit COICOP codes correctly, we need to choose a method of measure. For classifying $\boldsymbol{x}$ to a tree, I will use the term of agreement or not as a count of match. A match on a root term will give a higher score of $\boldsymbol{\alpha}$, than a match on any branches: $\boldsymbol{\beta}$. By scoring matches, we can better choose the correct tree. In the case of mapping item descriptions to 5-digit COICOP codes, ambiguity in classification can be blamed on the sparsity of data. In the ideal case, we would have all item descriptions and we would only need to map item descriptions directly to an existing tree.

4.1 Match for 5-digit COICOP code

For an indisputable choice of 5-digit COICOP codes, we require a match between terms of an item description $\boldsymbol{x}$ and $\vec{\boldsymbol{x}} \in \vec{\Omega}$

$$\gamma(\boldsymbol{x}, \vec{\boldsymbol{x}}) = \alpha \mathbb{I}_{root}(\boldsymbol{x}, \vec{\boldsymbol{x}}) + \sum_{k=1}^{K} \Big(\beta \mathbb{I}_{k,1}(\boldsymbol{x}, \vec{\boldsymbol{x}}) + \beta \mathbb{I}_{k,2}(\boldsymbol{x}, \vec{\boldsymbol{x}}) \Big) \tag{3}$$

where $\mathbb{I}_{k,1}$ indicates agreement in branch k preceding the root term and $\mathbb{I}_{k,2}$ for branch following the root term. K is the maximum number of branches in the entity forest.

We have three cases that comes from match on terms

1. *Full match on all terms:*

   We will have a direct agreement for all terms in x for item descriptions that already exists in our entity forest.

2. *Unique match:*

   Given a unique match on the root term. For instance ’blåkveite røkt’, the only root term is ’blåkveite’ within 5-digit COICOP code ’01.1.3.2’.

3. *Match on multiple trees*

   Match on multiple trees within the same 5-digit COICOP code, requires us to choose the tree with the highest match score $\gamma(\boldsymbol{x}, \vec{\boldsymbol{x}})$. For example, if we have a tie for item description ’vanilje is’, both ’vanilje’ and ’is’ are roots in 5-digit COICOP code ’$01.1.8.6$’. As both are roots within the same 5-digit COICOP, we do not need to look for further measures of branches.

In all these cases the 5-digit COICOP code is chosen without uncertainty.

4.2 Ties in Different 5-digit COICOP codes

In the case of ties between different 5-digit COICOP codes by $\gamma(\boldsymbol{x}, \vec{\boldsymbol{x}})$ we let

$$\psi_y(\boldsymbol{x}, \vec{\boldsymbol{x}}) = \sum_{\vec{\boldsymbol{x}} \in \vec{\Omega}_y} \mathbb{I}_{root}(\boldsymbol{x}, \vec{\boldsymbol{x}}) \phi(\boldsymbol{x}, \vec{\boldsymbol{x}})$$

where $\phi(\boldsymbol{x}, \vec{\boldsymbol{x}})$ measures the closeness of non-root terms given match on root term. When measuring ’closeness’ we can approach it in two different ways, as a measure of the number of instances in trees or by calculating closeness in terms used in $\boldsymbol{x}$ (non-root terms) and looking for cases of these terms in other trees within the same 5-digit COICOP code.

1. *Case 1*

   For each item description $\boldsymbol{x}_i$ in the original corpus Ω corresponds to one *root-path* in a tree for 5-digit COICOP code y_i, by which $\boldsymbol{x}_i$ is mapped to that root term. For an instance ’karamell sjokolade’, both terms exist in root($\vec{\Omega}$), ’sjokolade’ within 5-digit COICOP code ’$01.1.8.5$’ and ’karamell’ in 5-digit COICOP code ’$01.1.8.6$’. The item descriptions of the cases are given in Table 8 with the corresponding tree structure illustrated in 12.

   $$\text{root}\big(t(\boldsymbol{x}_i)\big) = \begin{cases} \text{karamell} & \text{if } i \in U_{01.1.8.6} \\ \text{sjokolade} & \text{if } i \in U_{01.1.8.5} \end{cases}$$

   For 5-digit COICOP code ’$01.1.8.5$’ we have three instances of the term ’sjokolade’ followed by ’karamell’, wheras only one instance within 5-digit COICOP code ’$01.1.8.6$’. The item description ’sjokolade karamell’ is placed in within ’$01.1.8.5$’.

Item	Item description	5-digit COICOP code
1	{sjokolade karamell karamell}	01.1.8.5
2	{sjokolade karamell sjoko}	01.1.8.5
3	{sjokolade karamell havsalt}	01.1.8.5
4	{dobbel sjokolade karamell}	01.1.8.6

Table 8: Excerpt from data for root term ’sjokolade’ and ’karamell’

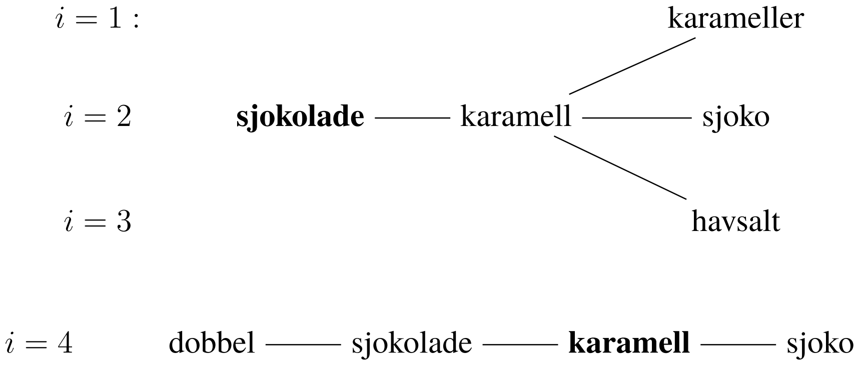


Figure 12: Extract from tree in ’01.1.8.5’ and ’01.1.8.6’

2. *Case 2*

   For ties where we only have a match on root term, and no match beyond in branches. For example, in the case of ’creme nougat pudding’; we have ’creme’ as the root term in 5-digit COICOP code ’01.1.8.6’, and ’nougat’ in 5-digit COICOP code ’01.1.8.5’. None of the terms in the item description has a case where they precede each other in the data. For the different 5-digit COICOP codes, we need to look at other trees where these words exist. For root-term "creme" within ’01.1.8.6’, we have three cases of ’nougat’, but non of ’pudding’. For root-term ’nougat’ within ’01.1.8.5’ we have seven cases of ’pudding’ in branches, but none of ’creme’. For ’creme nougat pudding’ we choose 5-digit COICOP code ’01.1.8.5’.

4.3 No Match on Root Term

Finally, in the case $t(\boldsymbol{x})$ has no matched root term in $\vec{\Omega}$ at all, let

$$\tau_y(\boldsymbol{x}) = \xi(\boldsymbol{x}, \vec{\boldsymbol{x}}_y)$$

be a measure of the closeness of a term in the item description in $t(\boldsymbol{x})$ to all the item descriptions $\vec{\boldsymbol{x}}$ within 5-digit COICOP codes, y. So for a new instance where no term exists in root($\vec{\Omega}$), we need to look for matches in branches. We score based on the number of branches and chose y for the case with the most trees with matches on a branch. In the case of ties, we approach it similarly to the ties on root terms. In the case of only branch matches, a higher level of uncertainty is prevalent.

4.4 No Match

If no match is found in any trees of the entity forest, this would be an appropriate step to introduce human-in-the-loop or a form for automatic classification as a possible solution. No match on root will happen in cases of an introduction of a new item or of misspellings of an already existing item and for these cases human-in-the-loop would be an option that may increase classification in the long run.

5 Results

Building an entity forest is a case of increasing purity of item descriptions by building trees based of root terms. When building our entity forest to classify items we would, ideally, like the entire dataset to be used. If all item descriptions for items in stores were available, classifying items to 5-digit COICOP codes would only be a case of mapping an instance to an existing tree. However, we do not have all item descriptions in the universe, and we need another way of measuring how well we can expect item descriptions to be classified to the correct 5-digit COICOP code. This means that we need a metric to measure the performance of our entity forest. Within machine learning, a commonly used metric is classification accuracy, defined as the proportion of correctly classified instances. We can calculate this metric by splitting the full dataset into two parts, typically referred to as the training set and test set [Raschka and Mirjalili, 2019]. In our case, the first partition is used to build the entity forest, while the other part is used to evaluate the performance using the data that was not used to build the entity forest. Choosing this metric to account for performance and compare it to previous works done by SSB may not be the best approach, but it is sufficient to prove that changing the way we look at data engineering is the way to go forward. An obvious issue that will arise is related to terms that have not been previously observed, which can prove challenging to classify. This will lead to a decrease in classification accuracy.

I have chosen to look at five different excerpts from the dataset, hereby denoted as case studies. By comparing the entity forest to works of Müller [2021] by using code written for his study, provided to me by SSB. From Müller [2021] the data is split into training data and test data, splitting so that 80% belongs to the training set and 20% to the test set. We compare the results obtained from his code to the building- and classifying of the entity forest, by respectively using 80% of the data to build the entity forest and 20% to be used in classification. Table 9 gives results from the following five case studies.

Case	Accuracy - Müller [2021]		Accuracy - Entity forest
1	0.8948905 (LR cv-ch23)	>	0.8928324
2	0.9569319 (LR cv-ch23)	<	0.9807692
3	0.9797579 (RF cv-ch23)	>	0.9706714
4	0.9199279 (LR cv-ch23)	<	0.9532026
5	0.9311064 (RF cv-ch23)	<	0.9597156

Table 9: Classification accuracy for the five different experiments on different data. Comparison between entity forest classification and the works of Müller [2021]

5.1 5-digit COICOP codes within a 4-digit COICOP code

5-digit COICOP codes that often contain similar terms and phrasings are the codes that appear within the same 4-digit COICOP code. We look at classification within two different 4-digit codes.

- **Case Study 1: 4-digit code '01.1.3'**

  We use 4-digit code: '01.1.3' - *Fish and other seafood*, which includes seven 5-digit codes:

  - '01.1.3.1' - Fish, live, fresh, chilled or frozen.
  - '01.1.3.2' - Fish, dried, salted, in brine or smoked.
  - '01.1.3.3' - Fish preparations.
  - '01.1.3.4' - Other seafood, live, fresh, chilled or frozen.
  - '01.1.3.5' - Other seafood, dried, salted, in brine or smoked.
  - '01.1.3.6' - Other seafood preparations.
  - '01.1.3.7' - Livers, roes and offal of fish and of other seafood in all forms.

  Because of the similarity issues these 5-digit codes are defined as hard to classify. For example;'torsk porsjon', 'torsk røkt' and 'torsk stekt' all belong to different 5-digit codes, respectively '01.1.3.1', '01.1.3.2' and '01.1.3.3'. Two or fewer instances exist of '01.1.3.5' and are therefore not taken into account in classification.

  Classification accuracy

For code proposed by Müller [2021], "LR cv-ch23" - Logistic Regression with the feature extractor CountVectorizer with (2,3) in N-gram range performs best with a classification accuracy of: 0.8948905.

When building an entity forest on 80 % of the data and classifying the remaining 20%, we receive a classification accuracy of 0.8928324. The differences are not significant, however, when looking closer at our misclassifications we see why classifying to the entity forest does not perform better.

Misclassified item descriptions from the entity forest

Taking a look at the misclassified item descriptions, we observe that 45% of misclassifications are due to the terms in item descriptions used to map onto trees does not exist as branches or roots. This really is a case of data sparsity. An excerpt of these include:

- koljekaker
- brosmefilet
- kveiteskiver
- hysefilet
- torskekarbonader
- fiskerogn
- krabbepostei
- klo
- bestemors grovkaker

The remaining 55% are due to:

1. Actual misclassifications. Some of these also occur because of data sparsity due to only building on 80% of the data, and we therefore lack some roots and some branches. An excerpt of these are given in Table 1:

Item description	Given Code	True Code
polarsnack filet	01.1.3.3	01.1.3.2
dorade hel fersk	01.1.3.3	01.1..3.1
bokna torsk	01.1.3.2	01.1.3.1
torskefilet	01.1.3.3	01.1.3.1
lutefisk filet vakuum	01.1.3.1	01.1.3.2
gravlaks skivet	01.1.3.3	01.1.3.2
brosmefilet	01.1.3.3	01.1.3.1

Table 10: Excerpt from misclassified items in 4-digit code 01.1.3

2. Entity forest built on wrongly labeled data. For example 'lettsaltet skiver' is by the entity forest classified as '01.1.3.2' - Fish, dried, salted, in brine or smoked. But is returned as misclassified as it is labeled as '01.1.3.1' - Fish, live, fresh, chilled or frozen, in the original dataset. As humans, we know that 'lettsaltet skiver' refers to treated food, not fresh, chilled, or frozen. Therefore, the classification done by the entity forest is the correct and true classification. An excerpt of different items where this is the case is given in Table 2.

Item description	Given and True Code	Labeled Code
lettsaltet skiver	01.1.3.2	01.1.3.1
maki laks dagsfersk	01.1.3.3	01.1.3.1
lofoten torsk hollandaise	01.1.3.3	01.1.3.2
brosme	01.1.3.1	01.1.3.4
panert rodspette	01.1.3.3	01.1.3.1
torskefilet lettsaltet	01.1.3.2	01.1.3.1

Table 11: Excerpt from wrongly misclassified items in 4-digit COICOP code 01.1.3

- **Case Study 2: 4-digit code '01.1.8'**

  Classification within a 4-digit code: '01.1.8' - *Sugar, confectionery and desserts*, which includes four 5-digit codes:

  - 01.1.8.1 - Cane and beet sugar.
  - 01.1.8.2 - Other sugar and sugar substitutes.
  - 01.1.8.3 - Jams, fruit jellies, marmalades, fruit purée and pastes, honey.
  - 01.1.8.4 - Nut purée, nut butter and nut pastes.

- 01.1.8.5 - Chocolate, cocoa, and cocoa-based food products.
- 01.1.8.6 - Ice, ice cream and sorbet.
- 01.1.8.9 - Other sugar confectionery and desserts

Classification accuracy

For code from Müller, 2021 the best-performing model is "LR cv-ch23" - Logistic Regression with the feature extractor CountVectorizer with (2,3) in N-gram range performs best with an accuracy of: 0.9569319.

When building an entity forest on 80 % of the data and classifying the remaining 20%, we get an accuracy of 0.9807692.

Misclassified item descriptions from the entity forest

Misclassification due to no terms being in the dataset is at 52%. This means that true misclassifications are due to the remaining 48%. Given that some of these also include wrongly misclassified items.

5.2 Randomized Choice of 5-digit codes.

We have 3 instances where we choose seven different 5-digit codes.

- **Case Study 3: First instance of seven random 5-digit codes.**
  - 01.1.3.1 - Fish, live, fresh, chilled or frozen.
  - 01.1.7.2 - Fruit-bearing vegetables, fresh or chilled.
  - 01.1.2.2 - Meat, fresh, chilled or frozen.
  - 01.1.4.8 - Eggs.
  - 01.2.1.0 - Fruit and vegetable juices
  - 01.1.8.5 - Chocolate, cocoa, and cocoa-based food products (
  - 01.1.6.2 - Citrus fruits, fresh

Models from Müller, 2021 give us the best performing model as ’RF cv-ch23’ - Random Forest with CountVectorizer as feature extractor and (2,3) in N-gram range, with a classification accuracy of 0.9797579.

Classification accuracy from building an entity forest from 80% of the data and mapping the remaining 20%: 0.9706714. 75% of misclassifications are because of no match on any root or branch.

- **Case Study 4: Second instance of seven random 5-digit codes.**
    - 01.1.9.1 - Ready-made food.
    - 01.1.9.9 - Other food products.
    - 02.1.3.0 - Beer.
    - 01.1.7.5 - Tubers, plantains and cooking bananas.
    - 01.1.2.4 - Offal, blood and other parts of slaughtered animals, fresh, chilled or frozen, dried, salted, in brine or smoked.
    - 01.1.4.3 - Other milk and cream,
    - 01.1.6.7 - Fruit, dried and dehydrated.

  For code from Müller, 2021 the best-performing model is "LR cv-ch23" - Logistic Regression with the feature extractor CountVectorizer with (2,3) in N-gram range performs best with an accuracy of: 0.9199279.

  Classification accuracy from building an entity forest from 80% of the data and mapping the remaining 20%: 0.9532026. 97% of misclassifications are due to no match on any root or branch.

- **Case Study 5: Third instance of seven random 5-digit codes.**
    - 01.1.8.6 - Ice, ice cream and sorbet.
    - 01.1.3.4 - Other seafood, live, fresh, chilled or frozen.

- 01.1.4.2 - Skimmed milk.
- 01.1.5.9 - Other animal oils and fats.
- 01.1.7.5 - Tubers, plantains and cooking bananas
- 01.1.6.9 - Fruit and nuts ground and other preparations.
- 01.2.6.0 - Soft drinks.

For code from Müller, 2021, the best performing model is 'RF cv-ch23' - Random Forest with CountVectorizer as feature extractor and (2,3) in N-gram range, with a classification accuracy of: 0.9311064.

Classification accuracy from building an entity forest from 80% of the data and mapping the remaining 20%: 0.9597156. 77 % of misclassifications are due to no match on any root or branch.

Increasing classification accuracy based on the results from splitting data into training and test should not be the goal, as this does not act as a train-test method in machine learning. Misclassification is largely due to data sparsity, leaving out 20% of the dataset. However, it further guides us in the direction of data exploration.

5.3 GitHub

The GitHub repository **COICOP-entity-forest** [Bauer-Nilsen, 2023] contains all the code used for preprocessing, building and classification. However, the datasets are not included in the repository as it belongs to SSB.

6 Discussion

In this thesis, I present a generative approach where I use maximum entropy classification and entity resolution as a framework for constructing the concept of an entity forest, to improve the classification of 5-digit COICOP codes. Where $D_{KL}(\mathbb{C}; \Omega)$ is maximized with respect to Ω during building and with respect to $\mathbb{C}$ during classification so that Maximum Entropy provides an elegant way to tie all these concepts together, which sets it apart from previous attempts made to COICOP classification.

The Discussion is split into four sections; Section 6.1 evaluates the results and some of the difficulties obtaining these from an entity forest. In Section 6.2 I discuss some limitations of the data used and other potential options in data engineering that were not implemented in this thesis. In Section 6.3 I discuss the potential limitations that arise from my choices regarding building and classification with the entity forest model. Section 6.4 presents the potential steps that can be taken that I believe will add a greater value to the entity forest model and its applicability in future work with COICOP datasets.

6.1 Evaluation and Results

Obtaining results with classification accuracy for the entity forest model as done in this thesis is a suboptimal approach. Since we leave out such a large percentage of the dataset a 100% classification accuracy seems almost impossible. Other approaches could be more optimal, however, I believe that other splitting techniques would yield similar results. For example, other common methods not used in this thesis that could have been implemented are; cross-validation and bootstrapping. These methods are often used to split data into training and test when training machine learning models, and could be used to evaluate the entity forest further. In the case of cross-validation, the choice of k, which refers to the number of folds, can influence model performance. If k is set too low, there may be a high variance in model performance due to the limited diversity of training data. This is particularly relevant in our case, where stratification and preserving group distribution in each fold are important for reliable evaluation. For instance, in our study, certain 5-

digit COICOP codes are limited in their representation, leading to few- and sparser trees in the entity forest. Within some 5-digit COICOP codes, the item descriptions share a high percentage of similar terms, whereas others do not. For example, we take a look at the different 5-digit codes within 4-digit COICOP code '01.1.3'- *Fish and other seafood*. From Table 12 we can see the percentage of unique words, the more unique words, the more different each item description is (in that they share fewer terms), which makes them potentially more difficult to classify.

5-digit COICOP code	**Unique terms (%)**	**n**	**n (%)**
01.1.3.1	30.27	258	9.05
01.1.3.2	25.54	382	13.39
01.1.3.3	18.61	1953	68.48
01.1.3.4	33.94	185	6.49
01.1.3.6	63.29	34	1.23
01.1.3.7	54.55	40	1.40

Table 12: Percentage of unique words in 5-digit codes within 4-digit code '01.1.3'

In Table 13, the 4-digit COICOP code '01.1.8' - *Sugar, confectionery and desserts* shows a slightly higher, but comparable, percentage of unique words. However, when examining the 5-digit COICOP code '01.1.8.5', which accounts for 99.5% of all item descriptions, we observe a significantly lower degree of uniqueness. This suggests that the terms within this code share many similar terms, making the item descriptions easier to classify. Given that the dataset is stratified on 5-digit codes we anticipate higher classification accuracy for '01.1.8', as it also appears in the results Table 9.

5-digit COICOP code	**Unique terms (%)**	**n**	**n (%)**
01.1.8.1	35.40	158	1.48
01.1.8.2	68.29	20	0.19
01.1.8.3	43.61	451	4.24
01.1.8.4	65.91	18	0.17
01.1.8.5	7.28	10607	99.49
01.1.8.6	35.74	437	4.11
01.1.8.9	49.22	464	4.36

Table 13: Percentage of unique words in 5-digit codes within 4-digit code '01.1.8'

This means that consideration of the choice of data partition and stratification is necessary to ensure robust evaluation of the classification. The choice of data used for building and classification can greatly impact the evaluation metrics. If the data used consists of easy-to-classify or challenging samples, it can artificially boost or underestimate the performance of our entity forest classification.

6.2 Preprocessing

In our case, data sparsity seems to be the true underlying problem regarding building an entity forest. As previously mentioned, if we were to have all possible item descriptions, a perfect mapping from item descriptions to existing trees would be possible. Choosing data will therefore significantly impact model performance and evaluation metrics. For the dataset used in this thesis, several options exist to improve the quality of item descriptions and the dataset as a whole. I chose not to look into the spellings of different item descriptions, but misspellings are bound to happen as the data is scanned in and written by consumers. This is also apparent in our data. Items not classified are often due to misspellings. Some examples are given in Table 14. Solving for misspellings by parsing or other techniques would increase classification accuracy when classified by splitting the dataset into building and classification. However, if all data was used to build trees and new instances appeared, the mapping would not take harm if some items were misspelled as long as one is correct.

Item description	Root that exists
sojaolje	soyaolje
bohvete	bokhvete
hetemel	hvetemel
balamicoeddik	balsamicoeddik
knekkebra	knekkebrød
kanelgriffel	kanelgiffel
macroni	macaroni
jorbær	jordbær

Table 14: Examples of misclassifications due to spelling mistakes.

Some 4-digit COICOP codes are also more difficult to classify than others, due to being (severely) misclassified. For example within 4-digit COICOP code '01.1.3' - Fish and other seafood, we have:

- 'kubbelys greige matt lakk'
- 'rosenberg grillpølse'

For these problems it seems easier to identify the misclassifications while reviewing the classification results. Therefore, a viable solution could involve implementing a continuous updating system designed to rectify misclassifications as they are identified either as an automatic solution or with an option of human-in-the-loop.

6.3 Building and Classification

The construction of the entity forest as proposed in this thesis primarily relies on rule-based techniques. However, it may be more suitable to use other building techniques. In my approach, I have chosen to place a greater emphasis on single terms as the defining characteristics of an item description, with bi-grams considered as a secondary option for the cases where single terms have already been used or are more frequently seen in other 5-digit codes. The frequency count is based on the total number of item descriptions rather than a percentage of occurrence in a 5-digit code. This can result in 5-digit codes with many item descriptions being favored over those with fewer item descriptions, even if the latter better represents a smaller 5-digit code.

When classifying item descriptions to existing trees I made the choice to weigh all branches equally because item descriptions do not necessarily follow the Norwegian language structure. This might however be a wrong deduction, as terms in branches closer to the root might have a larger significance than terms further away from the root. One solution is giving branches different scores of α_k and β_k by modifying 3, so that:

$$\gamma(\boldsymbol{x}, \vec{\boldsymbol{x}}) = \alpha_0 \mathbb{I}_{root}(\boldsymbol{x}, \vec{\boldsymbol{x}}) + \sum_{k=1}^{K} \Big(\alpha_k \mathbb{I}_{k,1}(\boldsymbol{x}, \vec{\boldsymbol{x}}) + \beta_k \mathbb{I}_{k,2}(\boldsymbol{x}, \vec{\boldsymbol{x}}) \Big)$$

where $\mathbb{I}_{k,1}$ indicates agreement (or not) at the sub-root level k preceding the root term and $\mathbb{I}_{k,2}$ that following the root terms, where K is the maximum number.

If there is no match when mapping an instance to a tree, I have suggested a solution that implements human-in-the-loop. As I have looked at classification by splitting the dataset, estimating the number of new instances with no similar terms when implementing the entity forest model for actual use is difficult. However, as shown in Section 5, half of the misclassified instances are due to terms being non-existent in the entity forest. By including a human-in-the-loop the misclassifications will be reduced by approximately 50%, or for Case Study 4 solve 97% of misclassified item descriptions.

6.4 The Way Forward

As mentioned in Section 6.3, other ways of building our forest might be more appropriate. A path that should be further researched is using supervised machine learning when building trees by finding fitting roots instead of a frequency measure as I have done in this thesis. A supervised machine learning model might better accommodate cases that include 5-digit codes with low representation and few terms. Machine learning can approach this by opting for a solution that allows for n-grams as root terms and uses stemming techniques from character-based methods. This can also increase the purity of roots- and improve building when adding in more 5-digit COICOP codes, as it is increasingly difficult to build when only allowing for single terms and bi-grams. For instance, 'sjokolademelk' and 'melk' belong to different 5-digit COICOP codes. Dividing 'sjokolademelk' into two terms, 'sjokolade' and 'melk,' would be counterproductive for Norwegian item descriptions, but creating a single term 'chocolate-milk' would be necessary for an English item description. On the other hand, dividing 'helmelk' and 'lettmelk' into 'hel', 'melk' and 'lett', 'melk' would improve the root purity of 'melk'. This task can be accomplished through supervised learning since such situations can be hard to handle in $\vec{\Omega}$ otherwise.

Another closely linked concept is that of a *rotating root*. I believe that this method requires little to implement, but can increase classification ability. For item descriptions having ’gravet laks’ and ’laks gravet’ as root terms, all these will refer to the same item belonging in 5-digit code ’01.1.3.2’. An excerpt of the trees with the two different root terms is given in Figure 13 and 14.

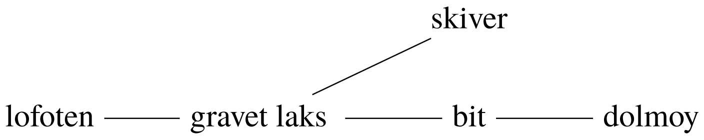


Figure 13: Extract from tree in ’01.1.3.2’ with root term ’gravet laks’

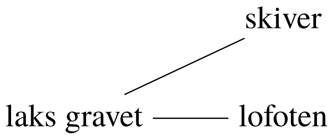


Figure 14: Extract from tree in ’01.1.3.2’ with root term ’laks gravet’

The concept of a *rotating root* implies that the root terms ’laks gravet’ and ’gravet laks’ are treated as one root term, without regard to the order, treated as interchangeable. Since item descriptions do not always strictly follow Norwegian language rules, including this as an option in n-gram roots can help to create more accurate mappings. This approach can prevent new instances from being misclassified and reduce the size of root, ultimately reducing classification time. An illustration of the new tree with a rotating root is shown in Figure 15.

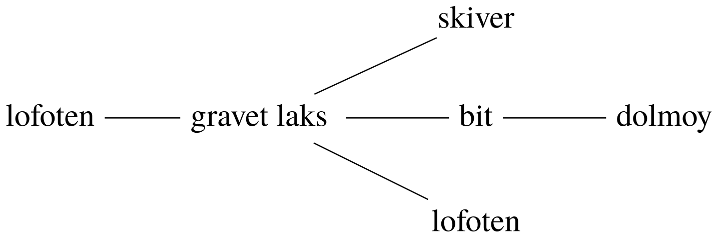


Figure 15: Tree in ’01.1.3.2’ with rotating root term ’gravet, laks’

7 Conclusion

In this thesis, I aimed to challenge the previous attempts of a COICOP classification method made by SSB to reduce manual work. I have argued that the large focus on modeling has been misguided, as data quality significantly affects the accuracy of AI-based classification. The concept proposed in this thesis introduces a way of transforming the corpus to construct an entity forest model by building trees from item descriptions. We observe that limitations regarding classification ambiguity that have been apparent in other studies on 5-digit COICOP classification can be improved by choosing a Maximum Entropy COICOP classification using entity forest. For cases of uncertainty, we have proposed human-in-the-loop as a solution. In conclusion, this thesis shows that looking at the COICOP classification from a different perspective displays the substantial potential of data engineering. The overall improvement in data quality offer several benefits beyond the scope of building an entity forest, extending to a range of other areas within official statistics.

References

Bayonne ham, Feb. 2022. URL https://en.wikipedia.org/w/index.php?title=Bayonne_ham&oldid=1125109940.

O. Abbott, P. Jones, and M. Ralphs. Large-scale linkage for total populations in official statistics. In *Methodological Developments in Data Linkage*, pages 170–200. John Wiley & Sons, Ltd, 2015. ISBN 978-1-119-07245-4. doi: 10.1002/9781119072454.ch8. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119072454.ch8.

L. Bauer-Nilsen. COICOP-entity-forest, 2023. URL https://github.com/Louiserb/COICOP-entity-forest.

L. Benedikt, C. Joshi, L. Nolan, N. Wolf, and B. Schouten. *Optical Character Recognition and Machine Learning Classification of Shopping Receipts*. Feb. 2020.

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996. URL https://aclanthology.org/J96-1002. Place: Cambridge, MA, Publisher: MIT Press.

S. Causevic. Generative vs Discriminative Probabilistic Graphical Models, Aug. 2021. URL https://towardsdatascience.com/generative-vs-2528de43a836.

Eurostat. Household budget survey - Microdata - Eurostat. URL https://ec.europa.eu/eurostat/web/microdata/household-budget-survey.

I. P. Fellegi and A. B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, Dec. 1969. ISSN 0162-1459. doi: 10.1080/01621459.1969.10501049. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1969.10501049.

K. Ganesan. What are Stop Words?, Apr. 2019. URL https://www.opinosis-analytics.com/knowledge-base/stop-words-explained/.

A. Holmøy and M. Lillegård. Forbruksundersøkelsen 2012 Dokumentasjonsrapport. Technical report, Mar. 2014. URL https://www.ssb.no/inntekt-og-forbruk/artikler-og-publikasjoner/_attachment/169278?_ts=144f30b31a8.

M. A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406): 414–420, 1989. ISSN 0162-1459. doi: 10.2307/2289924. URL https://www.jstor.org/stable/2289924.

D. Lee, L.-C. Zhang, and J. K. Kim. Maximum entropy classification for record linkage. *Survey Methodology*, 48(1):1–23, June 2021. ISSN 1492-0921.

D. M. Müller. Classification of consumer goods into 5-digit COICOP 2018 codes. Master's thesis, Norwegian University of Life Sciences, Ås, 2021. URL https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2981525.

A. Ng and M. Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf.

K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. Stockholom, Sweden, 1999.

S. Raschka and V. Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2, 3rd Edition*. Packt Publishing, 2019. ISBN 978-1-78995-575-0.

SSB. About Statistics Norway, Dec. 2022a. URL https://www.ssb.no/en/omssb/ssbs-virksomhet.

SSB. Survey of consumer expenditure, Dec. 2022b. URL https://www.ssb.no/en/inntekt-og-forbruk/forbruk/statistikk/forbruksundersokelsen.

UN. Classification of Individual Consumption According to Purpose (COICOP) 2018, 2018. URL `https://unstats.un.org/unsd/class/revisions/coicop_revision.asp`.

W. Winkler. Matching And Record Linkage. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6, Sept. 2014. ISSN 9780471598527. doi: 10.1002/wics.1317.

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway