# CaML: Carbon Footprinting of Household Products with Zero-Shot Semantic Text Similarity

Bharathan Balaji
bhabalaj@amazon.com
Amazon

Geoffrey Guest
gmg@amazon.com
Amazon

Venkata Sai Gargeya Vunnava
gvunnava@amazon.com
Amazon

Jared Kramer
jaredkra@amazon.com
Amazon

## ABSTRACT

Products contribute to carbon emissions in each phase of their life cycle, from manufacturing to disposal. Estimating the embodied carbon in products is a key step towards understanding their impact, and undertaking mitigation actions. Precise carbon attribution is challenging at scale, requiring both domain expertise and granular supply chain data. As a first-order approximation, standard reports use Economic Input-Output based Life Cycle Assessment (EIO-LCA) which estimates carbon emissions per dollar at an industry sector level using transactions between different parts of the economy. EIO-LCA models map products to an industry sector, and uses the corresponding carbon per dollar estimates to calculate the embodied carbon footprint of a product. An LCA expert needs to map each product to one of upwards of 1000 potential industry sectors. To reduce the annotation burden, the standard practice is to group products by categories, and map categories to their corresponding industry sector. We present CaML, an algorithm to automate EIO-LCA using semantic text similarity matching by leveraging the text descriptions of the product and the industry sector. CaML uses a pre-trained sentence transformer model to rank the top-5 matches, and asks a human to check if any of them are a good match. We annotated 40K products with non-experts. Our results reveal that pre-defined product categories are heterogeneous with respect to EIO-LCA industry sectors, and lead to a large mean absolute percentage error (MAPE) of 51% in $kgCO_2e$/\$. CaML outperforms the previous manually intensive method, yielding a MAPE of 22% with no domain labels (zero-shot). We compared annotations of a small sample of 210 products with LCA experts, and find that CaML accuracy is comparable to that of annotations by non-experts.

## CCS CONCEPTS

• **Information systems → Retrieval models and ranking**; • **Computing methodologies → Natural language processing**.

## KEYWORDS

carbon footprint, household products, EIO-LCA, CaML, NAICS


*WWW '23, April 30–May 4, 2023, Austin, TX, USA*

ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583882

**ACM Reference Format:**
Bharathan Balaji, Geoffrey Guest, Venkata Sai Gargeya Vunnava, and Jared Kramer. 2023. CaML: Carbon Footprinting of Household Products with Zero-Shot Semantic Text Similarity. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583882

## 1 INTRODUCTION

A rapid increase in greenhouse gas (GHG) emissions is warming our planet [33]. Household products and services contribute to 60% of these emissions [22]. Attribution of GHG emissions, measured in units of kilograms of carbon dioxide equivalent ($kgCO_2e$), to each product will drive awareness and change, from individual consumers to large corporations. Life Cycle Assessment (LCA) is a scientific framework that is used to estimate environmental $CO_2e$ emissions of products starting from raw material extraction to disposal or other end-of-life pathways [17, 21]. To precisely estimate the carbon embodied in a product, we need to know the materials and processes of manufacturing, transport data from manufacturer to customer, emissions during use such as fuel for a stove, and how the product is disposed. For global impact, we need to do such analysis on millions of products. We posit machine learning (ML) will play a crucial role in accelerating carbon attribution, LCA modeling, and decision-making for carbon abatement.

We take initial steps towards this vision by automating $CO_2e$ estimation of products at an industry sector level. Economic Input-Output (EIO) LCA reduces the effort involved by estimating the aggregate sector-level $CO_2e$ emissions based on the materials and energy use measured through economic transactions [41, 46]. To assign carbon emissions to a product, we need to find its corresponding industry sector defined by standards such as North American Industry Classification System (NAICS) [25] and International Standard Industrial Classification [42]. We use NAICS in our experiments, and consider products sold in the United States (US). The US government publishes an EIO-LCA database which provides the $kgCO_2e$ per dollar estimate for aggregated NAICS codes [19].

Picking from one of 1K+ NAICS codes is time consuming, and requires LCA expertise. To reduce annotation burden, carbon footprint reports aggregate products into categories, and then map the category to an industry sector [3, 26]. We consider the manual product category mapping based carbon emission estimates as our baseline. We minimize the annotation burden and increase the accuracy of product to NAICS mapping using natural language processing (NLP). Prior works have treated this as a supervised classification problem [40], where the product features are the input

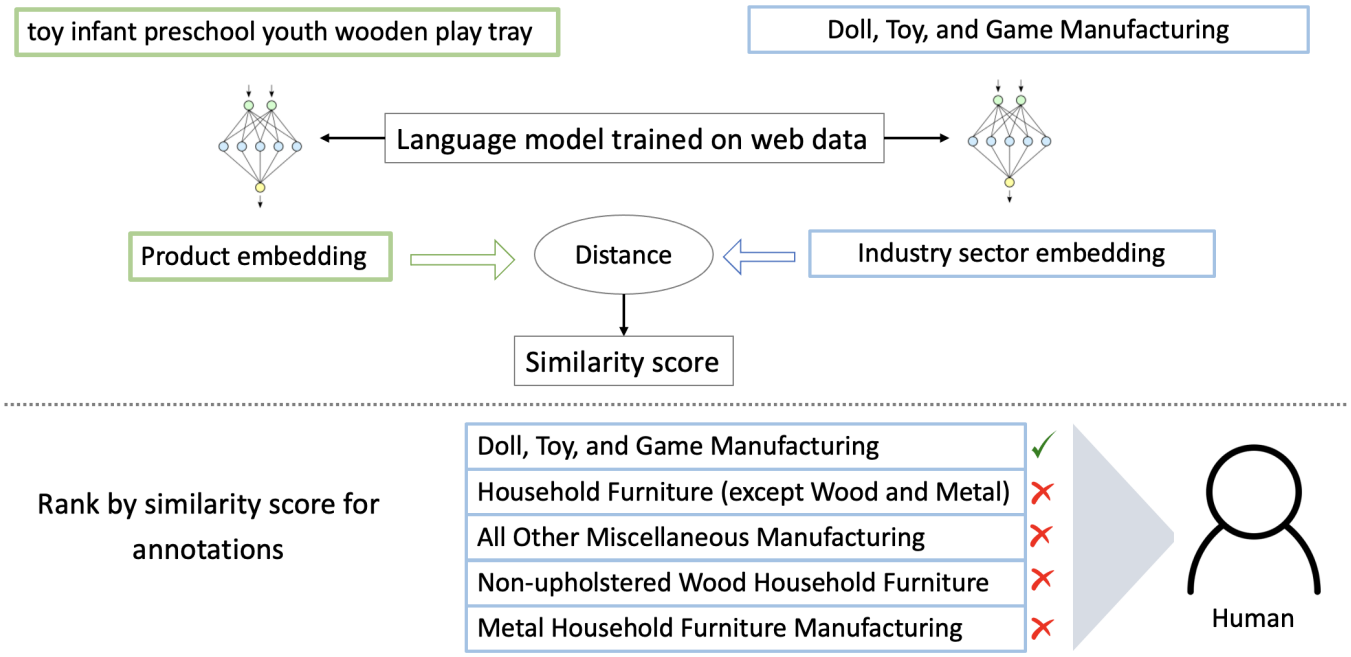


**Figure 1: CaML uses semantic text similarity to match products with industry codes and estimate $CO_2e$ emissions with EIO-LCA.**

and NAICS code is the output. However, the previous results were on hundreds of products, and need prohibitively large number of labelled examples for a model to generalize to millions of products. Instead, we leverage the text descriptions of the NAICS code and use semantic text similarity matching to identify an appropriate NAICS code for a product.

Our CaML (carbon assessment with ML) algorithm uses SBERT, a bi-encoder transformer model that generates a vector embedding given a sentence as an input, and uses cosine similarity between sentence embeddings as a measure of semantic similarity [35]. SBERT models are pre-trained on natural language inference datasets, and we use it as a zero-shot method to match products to NAICS codes. Given a product, we consider the top-5 matches ranked by cosine similarity and use it for annotations by a human. Choosing from five instead of 1K+ NAICS codes makes the annotation task much easier (~5x increase in labeling throughput). Human annotations help us both evaluate the zero-shot performance, and fine-tune the SBERT model for this specific domain. Figure 1 gives an overview of the CaML algorithm.

We consider products in the US retail sector for our experiments. We annotated 40K products using an annotation service, a dataset that is two orders of magnitude larger than prior state-of-the-art. We consider annotators as non-experts, and get consensus across 5 annotations per product. For our baseline, we use product categories defined by an e-commerce retailer. Treating the annotations as ground truth, we find that the baseline of mapping product category to NAICS codes gives an accuracy of 11% (a random classifier would give <0.1%). When we translate the NAICS codes to carbon emissions in units of $kgCO_2e/\$$ using EIO-LCA, the product category mappings give a mean absolute percentage error (MAPE) of 51%. The NAICS codes matches using the pre-trained SBERT model substantially outperforms the baseline, giving an accuracy of 48% (+3700bps) and a $kgCO_2e/\$$ MAPE of 22% (-2900 bps). We further annotated a random subset of 210 products with LCA experts, and found that non-experts are 46% accurate with a MAPE of 30%. CaML, on the other hand, gives an accuracy of 48% and MAPE of 25%. Therefore, the zero-shot CaML model is a scalable replacement for non-experts with about the same performance.

We open source our code and dataset with a permissive license[1].

## 2 BACKGROUND AND RELATED WORK

### 2.1 Life Cycle Assessment

LCA relies on life-cycle inventory, a dataset containing all relevant environmental, material and energy flows, to perform impact assessments such as global warming potential (GWP) and fresh water depletion. Our focus is on GWP, measured as carbon dioxide equivalent ($CO_2e$). There are two primary approaches to LCA: process-based and Economic Input-Output analysis-based (EIO) [10]. A process-based LCA is a bottom-up approach that tracks all the inputs (i.e. material and energy) and outputs (i.e. emissions and environmental wastes) of a product across its supply chain. The process LCA framework allows practitioners to dive deep into impacts of a specific product to identify hotspots in the supply chain. However, process-based LCAs are labor and time intensive, often requiring full tear-downs of the products.

EIO-LCAs take a top-down macroeconomic approach using supply-use tables provided by governments to estimate the emissions associated with the production of a unit currency worth of a

[1] https://github.com/amazon-science/carbon-assessment-with-ml

given good or service. An input-output matrix of industry sectors across the economy captures the inter-dependencies between the sectors as measured through economic transactions. The matrix quantifies how the demand in one sector impacts the rest of the economy. Environmental data such as water withdrawals, greenhouse gases, energy extraction, etc are collected for each sector in the economy, and are normalized into a unit of currency based on gross economic output by each sector [19]. This is an established method for estimating carbon emissions when detailed data are not available for a product. EIO is beneficial because it can be used to conduct LCAs using the accounting and financial data that companies already track. Leveraging these data sources removes the need to conduct an inventory of manufacturing a product, which shortens the time required for the analysis. Because we have a wide variety of products in the market, EIO-LCA can be used to identify the products that contribute most to environmental impacts, and target those for impact assessment deep dives.

The EIO-LCA dataset represents the financial transactions for the entire economy and are simplified by aggregating into industry sectors defined by economic codes such as North American Industry Classification System (NAICS) [25]. To identify the emissions associated with a product, we need to map it to one of these industry sectors. The government publishes the carbon emissions for each industry sector in units of $kgCO_2e/\$$, multiplying this factor by the sale price gives us the total carbon emissions of the product.

## 2.2 ML for LCA

ML has been recognized as a method to scale LCA in literature [2, 16]. Prior works have used ML in LCA for buildings [6], transportation [32], and various products [44]. We refer the reader to Algren et al. [2] for a survey. Froemelt et al. [14] used ML to cluster houses based on their environmental impact using a household consumption dataset. In contrast, we focus on environmental impact assessment of individual household products. Sousa and Wallace [40] proposed that ML can be used to create 'surrogate' LCAs for products that lack accurate ground truth information. They used neural networks to predict the energy consumption of 103 products with a maximum error of 40%. Wisthoff et al. [44] extended these ideas, and used ML for LCA of prospective design decisions that reduce environmental impact of 37 products. Other works have used ML to dive deeper into the supply chain of a single product, such as biochar [8] and sugarcane [30], to simulate the impact of design decisions. To our knowledge, we are the first to attempt EIO-LCA for products at scale. Our dataset is at least two orders of magnitude larger than prior works. Unlike prior methods that relied on simple supervision, we leverage label text for zero-shot prediction using language models trained on web data.

## 2.3 ML for NAICS code prediction

Prior works have also used ML for industry sector assignment based on text descriptions. Wood et al. [45] matched companies to NAICS codes using text data scraped from the web. They use bag-of-words, term frequency-inverse document frequency (tf-idf) for feature extraction, and a multi-layer perceptron for classification. The U.S. Bureau of Economic Census used write-in surveys to classify new businesses to specific NAICS codes using bag-of-words and logistic regression [13]. Oehlert et al. [31] use ML to validate NAICS codes reported in company tax forms using tf-idf and random forests. All of these works rely on a large labeled dataset for supervision. For example, Wood et al. [45] use 4 million labeled data points.

The Office of National Statistics in the UK use pre-trained embeddings to cluster companies and use Singular Value Decomposition (SVD) to extract descriptions for the cluster [38]. In contrast, our work matches products to existing NAICS codes with minimal labeling. Use of pre-trained language models also helps us support text from multiple languages, called out as a challenge in prior works. We focus on matching products to NAICS codes instead of companies because a single company can manufacture products that belong to multiple NAICS codes, e.g., Adidas manufactures both sports gear and apparel. Working with individual products gives us a more precise estimate of carbon emissions.

## 2.4 Natural language models

We use pre-trained SBERT models for encoding our text features [35]. SBERT has been used in a similar fashion for applications such as fact-checking [24], cyberbullying detection [15], and author representation [36]. Our algorithm is similar to the label embeddings proposed by Zhang et al. [48]. We do not claim novelty in NLP algorithms, and instead demonstrate that application of state-of-the-art in NLP can lead to step change in performance in the domain of LCA. While our results are promising, it can be potentially improved by exploiting the hierarchical structure of our labels [29, 37], or framing the problem as an entailment task [28]. We hope our work leads to further research into challenging tasks in the domain, such as automation of process-based LCA which requires extraction of bill-of-materials of products [5], and inferring environmental impacts from product disclosure documents [20].

# 3 DATASET AND CARBON ATTRIBUTION

We have selected the NAICS codes as the primary unit to which a product will be mapped to. NAICS is published by the US Census Bureau [25] and the commodities (products and services) in each NAICS code represent detailed resolution of industry classification. A single NAICS code can contain multiple industries. The 6-digit NAICS code are organized hierarchically, with 4-digit and 2-digit NAICS codes forming two levels in the hierarchy. Figure 3 in Appendix illustrates an example.

These NAICS codes are further translated to another set of economic codes called Bureau of Economic Analysis codes [1]. BEA codes aggregate multiple NAICS codes into higher level industry descriptions, which typically map to the corresponding 4-digit NAICS code. For example, the NAICS codes for chocolate and candy making (311320), granulated sugar production (311313), and crystallized fruits making (311340) all map to the BEA code for sugar and confectionery product manufacturing (311300). These BEA codes can then be used to calculate carbon emissions using the USEEIO (US Environmentally Extended Input Output) model published by the EPA [19]. The USEEIO model assigns a carbon emissions number with units of kg $CO_2$ equivalents per \$1 ($kgCO_2e/\$$) of commodities produced by the industries in a BEA code. If the product being mapped is an artisanal chocolate bar costing \$5, then it is first mapped to the NAICS code for chocolate making (311320), and then

translated into the BEA code for sugar/confectionery production (311300), and is finally assigned 5.7 $kgCO_2e$ (1.14 $kgCO_2e$\$ x \$5). We use the NAICS data available from `https://naics.com/` and carbon emission data published by Yang et al. [46].

We use products sold in the US from an e-commerce retailer. We create two datasets - one containing 40000 products, and another with 6646 food products for our experiments. The food dataset represents products from a variety of industry sectors to evaluate our method. In comparison, sectors such as books have minimal variations. For product description, we use a concatenation of the following texts from the product web page: title, description, and bullet points that describe additional attributes.

## 4 DATA PRE-PROCESSING

We clean the product and NAICS text descriptions using the Natural Language Toolkit [27]. We convert the text to lower case, and replace punctuation with underscores. As our descriptions are in English, we remove stopwords such as 'a', 'the', 'and'. We use lemmatization, and remove extra spaces, numbers, and special characters like parentheses. We remove repeating words such as 'manufacturing' and 'production' from NAICS text descriptions as they do not aid in classification. While such text cleaning procedures are not strictly required for natural language processing (NLP) models like SBERT, we find that cleaning the text leads to improvement in performance. We hypothesize that product descriptions often include special characters and formatting that impacts model performance. We present the impact of text pre-processing on model performance with an ablation analysis in Section 7.3.

## 5 CAML METHODOLOGY

CaML uses SBERT for semantic text similarity based matching [35]. SBERT, short for Siamese-BERT, is a bi-encoder transformer architecture that takes tokenized text as input and outputs a fixed-size embedding. SBERT is based on BERT [12] models that were trained with masked language model pre-training objective, and produced state-of-the-art NLP results at the time of release. We use SBERT models as they are computationally more efficient for sentence similarity task compared to BERT cross-encoder models. We use the "all-mpnet-base-v2" model from the sentence-transformer library because it gives the best average performance across sentence similarity benchmarks [18]. The model has been trained on 1 billion sentence pairs from a collection of 33 NLP datasets. The model consists of 110M parameters, is of size 420MB, and outputs 768 dimensional embeddings. We use a sentence length of 128 tokens, which translates to about 100 English words; the rest of the sentence is truncated. We include an ablation analysis with different ML models and sentence lengths in the Section 7.3.

NAICS codes have multiple industries associated with them even at the lowest level in the hierarchy (see Figure 3 in Appendix). CaML treats each industry description as a separate sentence in our semantic matching algorithm. The specificity of the NAICS description helps find improved matches with the SBERT model. In total, we have 11623 NAICS sentences. We observe the best performance when we concatenate the detailed industry descriptions with their corresponding BEA title, as we show in Section 7.3.

**Algorithm 1:** Pseudocode for CaML

```
Input  : product_text, naics_text_list, eio_lca_table
Output : product_kgCO2e_per_dollar
similarity_scores = []
product_text = preprocess(product_text)
product_embedding = model(product_text)
for naics_text in naics_text_list do
    naics_text = preprocess(naics_text)
    naics_embedding = model(naics_text)
    similarity_scores.append(
    cosine_similarity(naics_embedding, product_embedding))
naics_index = arg_max(similarity_scores)
naics_match = naics_text_list[naics_index]
product_kgCO2e_per_dollar = eio_lca_table(naics_match)
```

Algorithm 1 gives an overview of CaML. Both the product text and NAICS text are fed as inputs to the SBERT model after preprocessing to get their corresponding embeddings. CaML computes the cosine similarity of the product embedding and all the NAICS embeddings, and picks the NAICS code corresponding to the embedding with the highest similarity score as the best match. The EIO-LCA table consists of $kgCO_2e$/\$ values for each NAICS code. A lookup from this table gives us the product $kgCO_2e$/\$.

## 6 ANNOTATIONS AND METRICS

Given a product embedding, CaML ranks the NAICS sentences by the cosine similarity score of their embeddings. The top matches of NAICS sentences typically correspond to the same NAICS code. Therefore, CaML considers the top-20 matches, and aggregates them by their NAICS codes. After aggregation, we consider the top-5 NAICS codes ranked by decreasing cosine similarity for annotations. It is possible that the top-20 NAICS sentence matches yield less than 5 unique NAICS codes, we use them as-is for annotations. Figure 4 shows an example of our annotation task for an artisanal chocolate in our annotation interface after finding the top NAICS codes using the CaML algorithm. The artisanal chocolate has 10 NAICS sentence matches that belong to the same NAICS code in the top-20 matches. CaML aggregates the top-20 NAICS sentences to get 3 unique NAICS codes. We also include an equivalent Jupyter notebook based annotation interface in Figure 4 of Appendix B.

We use an annotation service for labeling the data at scale. Given the product description, an annotator can mark if each of the NAICS codes are a 'Match' with a checkbox, which we translate to a label of 1 or 0. It is possible that there is more than one match or none at all. The annotator can choose to skip products if unsure. They can also look up the product on the web, and verify the description of NAICS code online. We categorize our annotators as non-experts (crowd-sourced workers in the annotation service).

We also annotate a small sample of products with LCA scientists, and refer to them as expert annotations. An expert uses their experience of interpreting the constituents of the supply chain of a product when assigning a NAICS code. When they compare the product and NAICS description for annotations, they estimate the possible upstream industries that make up the supply chain of the product. E.g., for a chocolate drink some of the possible NAICS codes are 'chocolate liquor', and 'chocolate milk'. An LCA expert

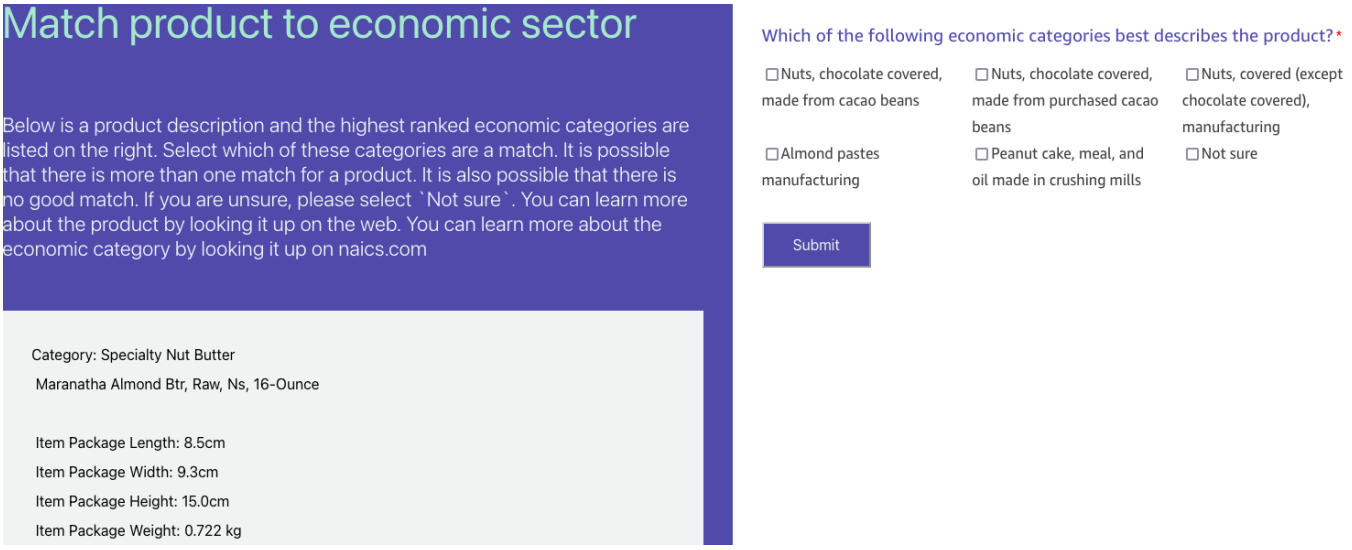


**Figure 2: Screenshot of the annotation task from Amazon SageMaker GroundTruth. We provide product and NAICS text descriptions to the SBERT model and find the top NAICS code matches for a product using cosine similarity. A crowd-source worker labels if the top ranked NAICS codes match the given product description. It is possible that there is more than one match or none at all.**

knows from their experience that such drinks have a milk component in their supply chains and they map the product to 'chocolate milk' without hesitation but a non-expert may get confused between the two mappings.

We performed a pilot experiment with 10 products across three pools of workers, with associated costs of \$0.012 (low complexity), \$0.024 (medium complexity) and \$0.36 (high complexity) respectively. Each product received three annotations to infer a consensus among workers. We found that the accuracy of annotations were similar regardless of the costs. We chose the medium complexity worker pool for larger scale experiments as their throughput (annotations per second) was the highest. We gather 5 annotations per product for larger experiments to increase fidelity.

For evaluating model performance, we only consider the top match ranked by cosine similarity. We consider a NAICS code prediction as correct if it is marked as a 'Match' by the annotator. In many cases, the NAICS codes marked by the expert, and the one matched by the model may be similar. For example, an expert marked a product as 'soft drink manufacturing', whereas the model matched it to 'bottled water manufacturing'. Both the NAICS codes map to the same carbon emission factor obtained from the EIO-LCA database. Therefore, we also measure the regression error in the estimation of carbon emissions in terms of $kgCO_2e/\$$.

We perform all our experiments on a p3.2x instance in Amazon Web Services, which contains an NVIDIA V100 Tensor Core GPU [4]. Model inference time with this instance is 0.55 ms on average. It takes an average of 117.4 ms to compute the best NAICS match for a product using cosine similarity. We have made our code available in the supplementary material.

## 7 RESULTS

We collect three sets of annotations with 210, 6646, and 40000 products respectively. The datasets follow a long-tailed distribution. There are 519 unique NAICS codes in the 40K dataset, where 20% of the codes accounts for 77% of the products. Other datasets follow a similar distribution. Table 1 summarizes our results.

### 7.1 Deep Dive with Small Dataset

We start our analysis from the 210 food products annotated by both experts and non-experts. The scale of annotation is small as we have access to only a few experts. We annotated a total of 288 food products with experts, of which only 210 had a match in the top-5 predictions by the model. Therefore, the recall for this dataset is 73%. We did not collect the expert ground truth for products with no good matches, and use the remaining 210 products for further analysis.

*7.1.1 Human Annotation Performance.* We have 5 non-expert annotations per product, and we only consider products for which at least two annotators agree. We pick the NAICS code that receives the highest number of votes as a 'Match', breaking the tie randomly when multiple NAICS codes received two votes each. We refer to the corresponding dataset as Non-expert $\geq$2/5. 206 of the food products we consider received at least two votes per NAICS code matched. We also report the result with a variation of the dataset where we consider only those products for which a single NAICS code received more than three votes. However, it reduces the number of eligible products to 132 (Non-expert $\geq$3/5).

If we use the expert labels as ground truth, annotations from non-experts give an accuracy of 46% for NAICS codes with $\geq$2/5 votes (random classifier gives <0.1% accuracy). In terms of predicting $kgCO_2e/\$$, the corresponding mean absolute percentage error

| # Products | Ground Truth | Predictor | Accuracy | MAPE | R2 | # NAICS |
|---|---|---|---|---|---|---|
| **Human-level performance: Expert vs Non-expert annotations** | | | | | | |
| 206 | Expert | Non-expert (>2/5 votes) | 46% | 30% | 0.21 | 63 |
| 132 | Expert | Non-expert (>3/5 votes) | 54% | 29% | 0.43 | 63 |
| 134 | Expert 1 | Expert 2 | 49% | 13% | 0.44 | 35 |
| **Baseline: Product category vs individual product mapping by humans** | | | | | | |
| 27708 | Non-expert (≥2/5 votes) | Category mapping | 11% | 51% | -0.22 | 503 |
| 16960 | Non-expert (≥3/5 votes) | Category mapping | 12% | 49% | -0.24 | 472 |
| 4591 | Non-expert (≥2/5 votes) | Category mapping | 17% | 40% | -0.62 | 244 |
| 2817 | Non-expert (≥3/5 votes) | Category mapping | 20% | 40% | -0.61 | 206 |
| 159 | Expert | Category mapping | 28% | 31% | -0.47 | 58 |
| **Proposed: Zero-shot CaML model vs human annotations** | | | | | | |
| 38218 | Non-expert (≥2/5 votes) | Zero-shot Model | 48% | 22% | 0.45 | 519 |
| 23283 | Non-expert (≥3/5 votes) | Zero-shot Model | 54% | 19% | 0.53 | 497 |
| 6318 | Non-expert (≥2/5 votes) | Zero-shot Model | 67% | 12% | 0.57 | 260 |
| 3879 | Non-expert (≥3/5 votes) | Zero-shot Model | 78% | 8% | 0.75 | 225 |
| 210 | Expert | Zero-shot Model | 48% | 25% | 0.34 | 63 |
| **Proposed: Fine-tuned CaML model vs human annotations – 4-fold cross-validation results** | | | | | | |
| 6318 | Non-expert (≥2/5 votes) | Fine-tuned Model | 52% | 22% | 0.19 | 260 |
| 3879 | Non-expert (≥3/5 votes) | Fine-tuned Model | 58% | 20% | 0.19 | 225 |
| 210 | Expert | Fine-tuned Model | 63% | 20% | 0.45 | 63 |

**Table 1: Summary of experiment results. We treat the NAICS code from the 'Ground Truth' column as the ground truth and compute metrics against the predicted NAICS codes in the 'Predictor' column. Accuracy measures the correctness of the NAICS code prediction, MAPE and R2 measure the mean absolute percentage error and coefficient of determination with respect to kgCO2e per dollar.**

(MAPE) is 30% with a coefficient of determination ($R^2$) of 0.21. The accuracy improves to 54% if we use ≥3/5 votes. The relatively low accuracy shows the difficulty of the task, where the annotator needs to pick from closely related NAICS codes. Even if we consider annotations by two experts on the same subset of products, we get an accuracy of only 49%, albeit with an improved MAPE of 13%. Prior works on NAICS classification report a similar problem with errors in manual labeling, with a dataset of labeled food related NAICS codes giving only 42% accuracy [45]. Therefore, we consider a model that predicts at ∼50% or higher accuracy as having human-level performance.

Krippendorff's Alpha is a standard measure of inter-annotator agreement, measured on a scale of 0 to 1 (0 is perfect disagreement, 1 is perfect agreement). The Krippendorff's Alpha for our dataset is 0.31 for the 6K food dataset and 0.25 for the 40K generic products dataset. We expect the agreement to be low given the large number of labels (519 unique NAICS codes) and workers (364). Similar values have been reported in literature for heavy-tailed distributions [23].

*7.1.2 Baseline: Product Category Mapping.* Our baseline is a mapping of product categories to NAICS codes by experts. The product categories are defined by an e-commerce service. The categories are fine-grained, with >5K of them in our dataset. Examples of product categories include 'berries', 'clipboards', 'fitness accessories', 'envelopes'. To avoid annotation of each product to a NAICS code, it is common practice to use a mapping of product category to a NAICS code for carbon footprint reports [3]. However, the categories are designed for a wide-variety of use cases, and may not align with the NAICS code definitions. For example, an 'earpiece headset' has the product category 'fitness accessories' but the corresponding NAICS code is 'telephone apparatus manufacturing'. Of 210 products annotated by experts, we have product category to NAICS code mapping for 159 of them. Considering the expert annotations as ground truth, the product category mapping gives an accuracy of 28%, MAPE of 31% and $R^2$ of −0.47. As the set of products that have both the category mapping and the annotations is a reduced sample (159 as opposed to 210), this is not a fair comparison. Therefore, we include results with the same set of products in Table 8 of Appendix D for both the small and the larger datasets. The results only change marginally, and our conclusions remain the same.

*7.1.3 CaML Performance.* We evaluate CaML predictions with the SBERT pre-trained model. For the 210 food products labeled by experts, the model yields an accuracy of 48%, with a MAPE of 25% with respect to $kgCO_2e/\$$. Therefore, the zero-shot CaML model is far superior to our baseline, and marginally exceeds the performance of non-expert annotations. We manually analyzed the errors made by the model in the expert dataset. About 15% of them were caused by mislabeling by the LCA expert. 30% of the predictions were similar to the NAICS code chosen by the LCA expert (the first four digits of the NAICS code matched), whereas 55% of the errors were not close matches. The errors are primarily caused by words in the product description that confused the model. For example, one of the products was a 'cake icing coloring gel', and the model labelled it as 'cake frosting manufacturing' whereas the LCA expert picked 'food coloring, synthetic, manufacturing'.

We fine-tuned the model with 4-fold cross-validation. The performance improves to 63% accuracy and 20% MAPE for the expert dataset. We use the correct NAICS code (as per expert annotation) among the top-5 ranked by the model as the positive example, and the other four codes as hard negatives. We use cross-validation as the 210 products dataset is small relative to typical ML datasets, and we want to evaluate the performance across all the datapoints. We experimented with different hyper-parameters manually in one of the four folds to improve performance: the number of epochs, the length of sentence, and different ways to clean text. The model performance remained stable with changes to hyper-parameters, with a change in accuracy of <2%. We report the results from the best of these hyper-parameters after evaluation on all four folds, and use the same hyper-parameters for the other datasets. We list the ranges of the hyper-parameters we tuned, and their final values in Appendix C.

## 7.2 Medium to Large Datasets

The trends we observe in the small dataset continue as we expand to thousands of products. After filtering out erroneous annotations, we get a total of 6318 annotated food products with ≥2/5 votes. The baseline of product category mapping gives an accuracy of 17%. CaML zero-shot model, on the other hand, gives an improved accuracy of 67% with a corresponding MAPE of 12%. Therefore, CaML again substantially outperforms the baseline. We get 3879 products with ≥3/5 votes, and the conclusion from the results remain the same. CaML gets an accuracy of 78% while the baseline gets 20%. The annotators have an option to pick 'No good match' in the interface, and they do not find a match for 1.5% of the products with ≥2/5 votes, i.e., our model recall is 98.5%. The improved recall compared to the smaller dataset with expert annotations reveals a gap in domain knowledge when we crowd-source labels.

Surprisingly, CaML performance dropped after fine-tuning. The cross-validation accuracy is 52% for the ≥2/5 and 58% for the ≥3/5 dataset respectively. We hypothesize that the performance of the model saturates at ∼50% accuracy due to noisy annotations by non-experts. To improve accuracy, we would need ground truth labels that can be treated as a gold standard. Future works can endeavour to learn models that correct for noisy labels [9], and improve annotation quality with better interface design [11].

Our results generalize beyond food products. We evaluate CaML on a dataset of 40000 products from the US retail sector, going beyond food items to include clothing, electronics, pharmacy, automotive and more. We get 38218 products with ≥2/5 votes, and a corresponding product category mapping. CaML zero-shot model achieves a recall of 99.7% and an accuracy of 48% compared to 11% with our baseline. The resulting carbon estimate gives a MAPE of 22% for $kgCO_2e/\$$ estimation. The results are similar if we consider ≥3/5 votes (Table 1), or control for the same number of products (Table 8 in Appendix).

## 7.3 Ablations

We perform an ablation analysis to quantify the impact of our design decisions. All the results are based on the large 40000 products dataset. We start with variations of text pre-processing, summarized in Table 2. If we do no pre-processing and use raw text as model input, the performance drops dramatically (-20.9% in accuracy, +14.5% in MAPE). If we do not remove English stop words or common words in NAICS description then accuracy drops by >5%, although the impact to regression metrics is lower. Other pre-processing steps have minor impact.

| Method | Accuracy | MAPE | R2 |
|---|---|---|---|
| No pre-processing | 27.3% | 37% | -0.005 |
| Keep numbers | 44.2% | 25.9% | 0.38 |
| Keep stop words | 36.1% | 29.3% | 0.29 |
| No lemmatize | 43.2% | 24.8% | 0.38 |
| Include common words | 35.8% | 28.3% | 0.21 |
| **Default** | **48.2%** | **22.5%** | **0.45** |

**Table 2: Ablation of text pre-processing methods. Our default method in CaML removes numbers, removes English stop words, uses lemmatization, and removes common words in NAICS text such as 'manufacturing'.**

| Method | Accuracy | MAPE | R2 |
|---|---|---|---|
| BEA title only | NA | 49.2% | -0.02 |
| NAICS title only | 13.9% | 57.4% | -0.75 |
| NAICS + BEA title | 20.6% | 43.4% | -0.1 |
| NAICS description | 32.5% | 35.4% | -0.005 |
| **Default: NAICS description + BEA title** | **48.2%** | **22.5%** | **0.45** |

**Table 3: Ablation of NAICS text input. CaML uses a concatenation of NAICS industry description and BEA title.**

Next, we ablate the NAICS text that is used to create the label embedding (Table 3). The label that is directly attributed to the carbon emissions is BEA title. However, the title description is too vague to match with specific products. For example, the BEA title of 'Oilseed' includes a variety of products such as mustard, soybean, flaxseed, and canola. The SBERT models do not infer such hierarchical relationships and give a low cosine similarity score. This leads to poor performance, MAPE increases by 26.7% compared to our default method. As we increase the specificity of the text input, the performance improves. However, we see the best performance with a concatenation of detailed NAICS industry description and the BEA title, which captures the relationship between them.

We vary the sentence length of the inputs to the SBERT model, and found the performance improves with increasing length (Table 5). Performance is worst at 32 tokens with a 7% reduction in accuracy. The performance saturates at 128 tokens, which is the default choice for the rest of the results.

We measure the impact on performance by using different SBERT pre-trained models for zero-shot prediction of NAICS codes (Table 5). The performance is inline with the benchmark results published in `sbert.net`, with a decrease in performance as model size decreases. The 'all-*' models were trained on generic text with over 1 billion training pairs. The 'qa*' models were trained on question and answer sentence pairs and 'paraphrase*' models were trained

| Sentence length | Accuracy | MAPE | R2 |
|---|---|---|---|
| 32 | 41.2% | 27.7% | 0.31 |
| 64 | 44.8% | 24.8% | 0.4 |
| **Default: 128** | **48.2%** | **22.5%** | **0.45** |
| 256 | 48.2% | 22.5% | 0.45 |
| 512 | 48.2% | 22.6% | 0.45 |

**Table 4: Sensitivity to sentence length on CaML performance**

| Pre-trained Model | Acc | MAPE | R2 | Size |
|---|---|---|---|---|
| paraphrase-albert-small-v2 | 17.0% | 53.2% | -0.32 | 43MB |
| all-MiniLM-L12-v2 | 25.6% | 39.2% | -0.15 | 120MB |
| all-distilroberta-v1 | 21.6% | 44.4% | -0.41 | 290MB |
| multi-qa-mpnet-base-dot-v1 | 27.1% | 38.3% | -0.06 | 420MB |
| paraphrase-multi-mpnet-base-v2 | 20.9% | 44.5% | -0.07 | 970MB |
| **Default: all-mpnet-base-v2** | **48.2%** | **22.5%** | **0.45** | **420MB** |

**Table 5: CaML performance with different pre-trained SBERT models.**

| Model | Accuracy | MAPE |
|---|---|---|
| XGBoost + TF-IDF + Classification | 21.3% | – |
| XGBoost + TF-IDF + Regression | – | 52.9% |
| XGBoost + SBERT + Classification | 25.6% | – |
| XGBoost + SBERT + Regression | – | 45.7% |
| **Default: zero-shot CaML** | **49.4%** | **22.3%** |

**Table 6: Comparison of CaML zero-shot performance with fully supervised solution**

for paraphrase mining dataset. The generic 'all-*' models trained with the largest dataset yield the best performance. The 'mpnet' model outperforms other models of similar size due to an improved training objective that uses permuted language modeling instead of masked language modeling used by BERT [39]. The 'MiniLM' model outperforms 'distilroberta' with an distillation method [43].

We train fully supervised models on the 40000 product dataset and compare against CaML zero-shot performance, splitting the data by 3:1 for train and test. We use XGBoost models with default hyper-parameters, a well-known and robust algorithm [7]. We evaluate both TF-IDF vectorization [34] and SBERT embeddings to convert raw text into vectors. The results are poor, with less than half the performance of the zero-shot CaML model (Table 6). We ensure that at least 2 datapoints exist for each class. The number of NAICS code in the dataset drops from 519 to 444, indicating the large number of classes which only had one datapoint. The highly imbalanced nature of the dataset makes it challenging to learn a good supervised model with just 40,000 datapoints. The challenging nature of NAICS code classification has been observed by prior work. Even with 4 million labelled datapoints, Wood et al. [45] achieve a classification accuracy of only 47.9%.

## 8 CONCLUSIONS AND FUTURE WORK

We presented a semantic text similarity algorithm to estimate carbon emission of household products using EIO-LCA methods. Our algorithm matches a product to a corresponding industry sector for which there are published carbon emission factors in terms of $kgCO_2e/\$$. We annotated 40K products in the US retail sector, and found the zero-shot model predictions significantly outperforms manual mapping of product category to an economic sector. CaML zero-shot method also outperform direct supervision with a limited number of labelled examples. We annotated a small sample of products with LCA experts, and find that the model predictions are comparable to the annotations by a non-expert. These initial results are promising, and significantly reduce reliance on annotations compared to prior state-of-the-art methods based on supervision.

We focused on text based prediction of industry sectors. In some cases, the product text descriptions are ambiguous because of branding and keywords optimized for search engines. Use of product images with a multi-modal model can generate more appropriate embeddings. The hierarchy of NAICS codes can also be exploited to improve classification accuracy by encoding it as part of the loss function [47].

Although we mapped products to NAICS codes to complete the EIO-LCA, process-based LCAs face similar manual bottlenecks. For example, when conducting a process-based LCA, one must first identify the materials and manufacturing processes used to create a product. Then, each material or process must be assigned to the most appropriate environmental impact factor. Additionally, this work enables us to not only quantify greenhouse gas emissions of a product category, but also opens the door to include water, waste, biodiversity, and a host of other environmental and socially relevant impact categories in future LCAs.

We want to empower consumers to understand and reduce their own carbon footprints. The first enabling step in that direction is footprint estimates for the products. The effort to more accurately map hundreds of millions of products to the most appropriate environmental impact factor requires a scalable and accurate prediction algorithm. Use of machine learning is new in this domain. We have made initial strides towards estimating carbon emissions at scale. For downstream applications, we need to determine the uncertainties associated with the model predictions, and take the uncertainties into account in decision making.

While we have focused on estimating the overall carbon footprint, there is additional work required to compare two products and determine which of them have lesser carbon emissions. To make such decisions, we need to be more precise about which aspects of the product that impact the carbon footprint. Currently such work is done manually by experts, tools to automate or augment such decisions are an important direction of future work.

## REFERENCES

[1] 2022. Bureau of Economic Analysis. https://www.bea.gov/data
[2] Mikaela Algren, Wendy Fisher, and Amy E Landis. 2021. Machine learning in life cycle assessment. In *Data Science Applied to Sustainability Analysis*. Elsevier, 167–190.
[3] Amazon. 2021. Carbon methodology: Reaching net-zero carbon by 2040. Measuring, mapping, and reducing carbon the Amazonian Way. https://sustainability.aboutamazon.com/carbon-methodology.
[4] AWS. 2022. Amazon EC2 P3 Instances. https://aws.amazon.com/ec2/instance-types/p3/.

[5] Callie W Babbitt, Hema Madaka, Shahana Althaf, Barbara Kasulaitis, and Erinn G Ryen. 2020. Disassembly-based bill of materials data for consumer electronic products. *Scientific Data* 7, 1 (2020), 1–8.

[6] Natalia Nakamura Barros and Regina Coeli Ruschel. 2020. Machine learning for whole-building life cycle assessment: A systematic literature review. In *International Conference on Computing in Civil and Building Engineering*. Springer, 109–122.

[7] Tianqi Chen 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.

[8] Fangwei Cheng, Hongxi Luo, and Lisa M Colosi. 2020. Slow pyrolysis as a platform for negative emissions technology: An integration of machine learning models, life cycle assessment, and economic analysis. *Energy Conversion and Management* 223 (2020), 113258.

[9] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. 2020. Learning with Instance-Dependent Label Noise: A Sample Sieve Approach. In *International Conference on Learning Representations*.

[10] Mary Ann Curran. 2018. Michael Z. Hauschild, Ralph K. Rosenbaum, and Stig Irvin Olsen (eds): Life Cycle Assessment—Theory and Practice.

[11] Tobias Daudert. 2020. A web-based collaborative annotation and consolidation tool. In *Proceedings of the 12th Language Resources and Evaluation Conference*. 7053–7059.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[13] Brian Dumbacher and Anne Russell. 2019. *Using Machine Learning to Assign North American Industry Classification System Codes to Establishments Based on Business Description Write-Ins*. Technical Report. U.S. Census Bureau, Joint statistical meeting.

[14] Andreas Froemelt, David J Durrenmatt, and Stefanie Hellweg. 2018. Using data mining to assess environmental impacts of household consumption behaviors. *Environmental Science & Technology* 52, 15 (2018), 8467–8478.

[15] Oguzhan Gencoglu. 2020. Cyberbullying detection with fairness constraints. *IEEE Internet Computing* 25, 1 (2020), 20–29.

[16] Ali Ghoroghi, Yacine Rezgui, Ioan Petri, and Thomas Beach. 2022. Advances in application of machine learning to life cycle assessment: a literature review. *The International Journal of Life Cycle Assessment* (2022), 1–24.

[17] Michael Z Hauschild, Ralph K Rosenbaum, and Stig Irvin Olsen. 2018. *Life cycle assessment*. Vol. 2018. Springer.

[18] HuggingFace. 2022. all-mpnet-base-v2 sentence-transformer model. https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

[19] Wesley W Ingwersen, Mo Li, Ben Young, Jorge Vendries, and Catherine Birney. 2022. USEEIO v2. 0, The US Environmentally-Extended Input-Output Model v2. 0. *Scientific Data* 9, 1 (2022), 1–24.

[20] International Organization for Standardization. 2006. Environmental labels and declarations—Type III environmental declarations—Principles and procedures. ISO 14025.

[21] IIS ISO. 2006. ISO-14040 Environmental management–life cycle assessment–principles and framework: International Organization for Standardization. (2006).

[22] Diana Ivanova, Konstantin Stadler, Kjartan Steen-Olsen, Richard Wood, Gibran Vita, Arnold Tukker, and Edgar G Hertwich. 2016. Environmental impact assessment of household consumption. *Journal of Industrial Ecology* 20, 3 (2016), 526–536.

[23] Hamid Jalalzai, Pierre Colombo, Chloé Clavel, Éric Gaussier, Giovanna Varni, Emmanuel Vignon, and Anne Sabourin. 2020. Heavy-tailed representations, text polarity classification & data augmentation. *Advances in Neural Information Processing Systems* 33 (2020), 4295–4307.

[24] Neema Kotonya and Francesca Toni. 2020. Explainable Automated Fact-Checking for Public Health Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7740–7754.

[25] Jayanthi Krishnan and Eric Press. 2003. The north american industry classification system and its implications for accounting research. *Contemporary Accounting Research* 20, 4 (2003), 685–717.

[26] Murat Kucukvar, Nuri C Onat, Galal M Abdella, and Omer Tatari. 2019. Assessing regional and global environmental footprints and value added of the largest food producers in the world. *Resources, Conservation and Recycling* 144 (2019), 187–197.

[27] Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

[28] Bill MacCartney and Christopher D. Manning. 2008. Modeling Semantic Containment and Exclusion in Natural Language Inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Coling 2008 Organizing Committee, Manchester, UK, 521–528. https://aclanthology.org/C08-1066

[29] Taro Miyazaki, Kiminobu Makino, Yuka Takei, Hiroki Okamoto, and Jun Goto. 2019. Label embedding using hierarchical structure of labels for twitter classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 6317–6322.

[30] Sri Mursidah, Taufik Djatna, Anas Miftah Fauzi, et al. 2020. Supply Chain Sustainability Assessment System Based on Supervised Machine Learning Techniques: The Case for Sugarcane Agroindustry. In *2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA)*. IEEE, 1–7.

[31] Christine Oehlert, Evan Schulz, and Anne Parker. 2022. NAICS Code Prediction Using Supervised Methods. *Statistics and Public Policy* 9, 1 (2022), 58–66.

[32] Federico Perrotta, Tony Parry, Luís C Neves, and Mohammad Mesgarpour. 2018. A machine learning approach for the estimation of fuel consumption related to road pavement rolling resistance for large fleets of trucks. (2018).

[33] Hans O Portner, Debra C Roberts, Helen Adams, Carolina Adler, Paulina Aldunce, Elham Ali, Rawshan Ara Begum, Richard Betts, Rachel Bezner Kerr, Robbert Biesbroek, et al. 2022. Climate change 2022: impacts, adaptation and vulnerability. (2022).

[34] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. New Jersey, USA, 29–48.

[35] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[36] Rafael A Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning Universal Authorship Representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 913–919.

[37] Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical multi-label text classification using only class names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4239–4249.

[38] Michael Snow and Big Data Team. 2018. *Unsupervised document clustering with cluster topic identification*. Technical Report.

[39] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems* 33 (2020), 16857–16867.

[40] Ines Sousa and David Wallace. 2006. Product classification to support approximate life-cycle assessment of design concepts. *Technological Forecasting and Social Change* 73, 3 (2006), 228–249.

[41] Sangwon Suh. 2009. *Handbook of input-output economics in industrial ecology*. Vol. 23. Springer Science & Business Media.

[42] United Nations Statistical Division. 2008. *International Standard industrial classification of all economic activities (ISIC)*. Number 4. United Nations Publications.

[43] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[44] Addison Wisthoff, Vincenzo Ferrero, Tony Huynh, and Bryony DuPont. 2016. Quantifying the impact of sustainable product design decisions in the early design phase through machine learning. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 50145. American Society of Mechanical Engineers, V004T05A043.

[45] Sam Wood, Rohit Muthyala, Yi Jin, Yixing Qin, Nilaj Rukadikar, Amit Rai, and Hua Gao. 2017. Automated industry classification with deep learning. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 122–129.

[46] Yi Yang, Wesley W Ingwersen, Troy R Hawkins, Michael Srocka, and David E Meyer. 2017. USEEIO: A new and transparent United States environmentally-extended input-output model. *Journal of cleaner production* 158 (2017), 308–318.

[47] Hsiang-Fu Yu, Kai Zhong, Jiong Zhang, Wei-Cheng Chang, and Inderjit S Dhillon. 2022. PECOS: Prediction for Enormous and Correlated Output Spaces. *Journal of Machine Learning Research* (2022).

[48] Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. Multi-Task Label Embedding for Text Classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4545–4553.

# APPENDIX

## A HIERARCHY OF NAICS CODES

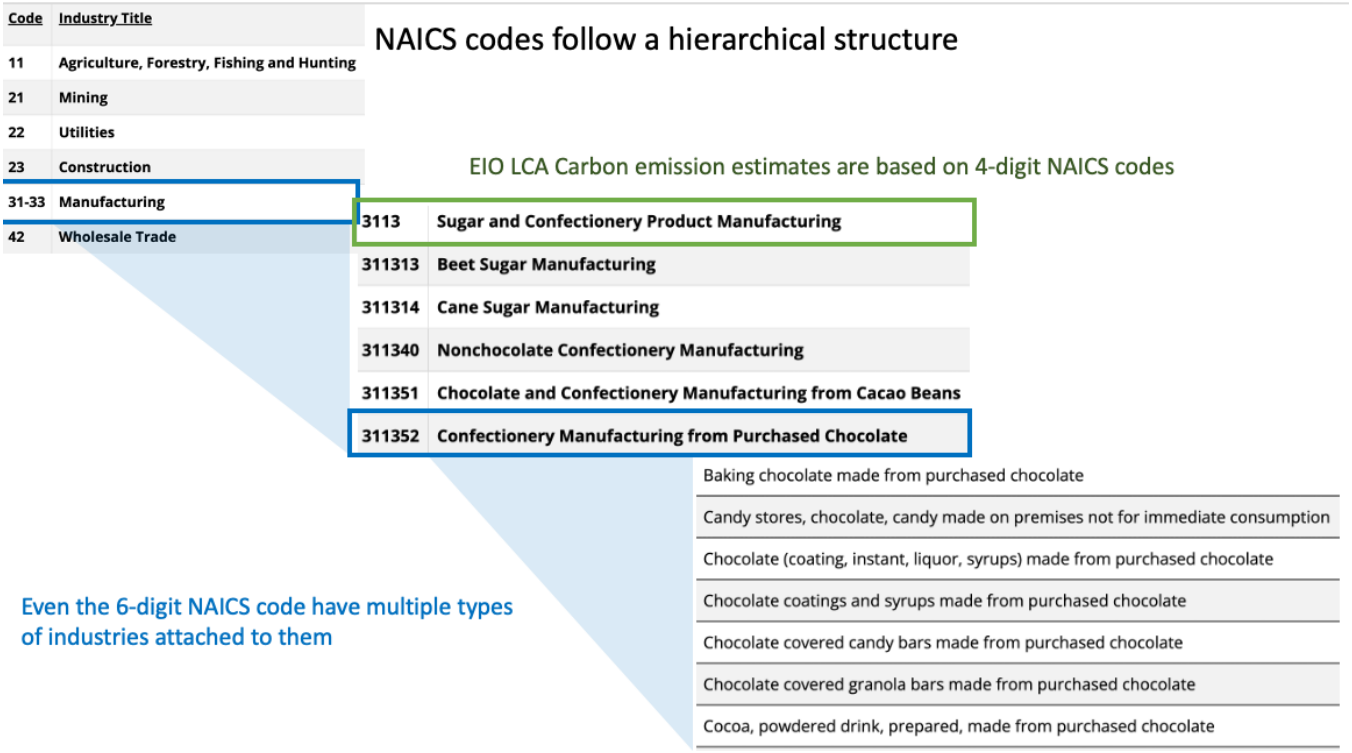


**Figure 3: NAICS codes represent economic sectors defined by the US government.**

## B ANNOTATION TASK

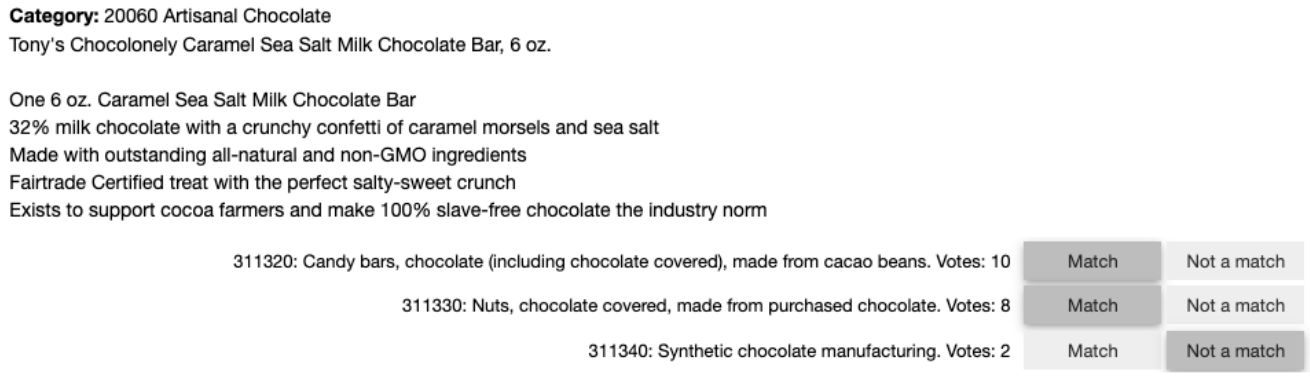


**Figure 4: Screenshot of the annotation task from our Jupyter notebook. We provide product and NAICS text descriptions to the SBERT model and find the top NAICS code matches for a product using cosine similarity. An annotator labels if the top ranked NAICS codes are a 'Match' or 'Not a match'. We use the annotations as labels for fine-tuning the SBERT model. It is possible that there is more than one match or none at all. In the example shown above, the first two options are good matches as they both describe a chocolate industry. It is difficult to tell from the product description if the manufacturer purchased the chocolate or made it directly from cacao beans.**

## C HYPER-PARAMETERS

We manually tuned the hyper-parameters in one of the four folds of our 210 product dataset. We provide the range of hyper-parameters we tuned, and the final values in Table 7. The rest of the hyper-parameters are the default values listed in `sbert.net`.

**Table 7: Hyper-parameters for fine-tuning the CaML model.**

| Hyper-parameters | Tuning Range | Final Value |
|---|---|---|
| Token length | {128, 256, 512} | 128 |
| Epochs | 4 - 10 | 5 |
| Warm up steps | – | 100 |
| Batch size | – | 16 |

## D RESULTS WITH SAME NUMBER OF PRODUCTS

**Table 8: Results where we control the number of products to be the same in both baseline and proposed solution. We treat the NAICS code from the 'Ground Truth' column as the ground truth and compute metrics against the predicted NAICS codes in the 'Predictor' column. Accuracy measures the correctness of the NAICS code prediction, MAPE and R2 measure the mean absolute percentage error and correlation of determination with respect to kgCO2e per dollar.**

| # Products | Ground Truth | Predictor | Accuracy | MAPE | R2 |
|---|---|---|---|---|---|
| **Human-level performance: Expert vs Non-expert annotations** | | | | | |
| 156 | Expert | Non-expert (≥2/5 votes) | 44% | 31% | 0.29 |
| 132 | Expert | Non-expert (≥3/5 votes) | 54% | 29% | 0.43 |
| 134 | Expert 1 | Expert 2 | 49% | 13% | 0.44 |
| **Baseline: Product category vs individual product mapping by humans** | | | | | |
| 27708 | Non-expert (≥2/5 votes) | Category mapping | 11% | 51% | -0.22 |
| 16960 | Non-expert (≥3/5 votes) | Category mapping | 12% | 49% | -0.24 |
| 4591 | Non-expert (≥2/5 votes) | Category mapping | 17% | 40% | -0.62 |
| 2817 | Non-expert (≥3/5 votes) | Category mapping | 20% | 40% | -0.61 |
| 156 | Expert | Category mapping | 27% | 32% | -0.47 |
| **Proposed: Zero-shot CaML model vs human annotations** | | | | | |
| 27708 | Non-expert (≥2/5 votes) | Zero-shot Model | 49% | 22% | 0.44 |
| 16960 | Non-expert (≥3/5 votes) | Zero-shot Model | 55% | 18% | 0.50 |
| 4591 | Non-expert (≥2/5 votes) | Zero-shot Model | 67% | 13% | 0.60 |
| 2817 | Non-expert (≥3/5 votes) | Zero-shot Model | 78% | 8% | 0.77 |
| 156 | Expert | Zero-shot Model | 49% | 25% | 0.40 |