Prepared by Deivyansh Singh

AI Models for Carbon Footprinting

Challenges and Future Trajectories of Models and CodeCarbon

Introduction

Carbon accounting has become a major concern for organisations in recent years. To address this, several models are available on the market. In this report, we will review three models—Parakeet, Flamingo, and CaML—used for recommendations on carbon footprinting. These models have revolutionised the intensive process of emission factor (EF) recommendations, which is a crucial component of life cycle assessment (LCA). The following report gives an idea of current limitations and development pathways based on the available research.

Challenges faced with each model

Parakeet Model

Parakeet Model aims to resolve the common problem of life cycle assessment (LCA), which is to find an appropriate emission factor from databases. Even though it has achieved great Precision@1 of 86.5%, it still has potential bottlenecks:

1. Domain-Specific Accuracy: It’s performance varies industry-wise, which requires special training for certain sectors.
2. Data: The model mostly depends on EF databases, which may have potential gaps from emerging technologies or materials.
3. Balance in Interpretation: Justifications that are technically accurate and accessible are a big challenge for non-experts to interpret.

CaML Model

CaML Model tackles the limitations of Economic Input-Output Life Cycle Assessment

(EIO-LCA). It is used for mapping products to potential sectors, reducing MAPE from 55% to 22%, but it still faces lots of challenges:

1. Semantic Ambiguity: Due to product descriptions with a lot of technicalities or ambiguous terminologies, it gets complicated for accurate sector mapping.
2. Granularity Tradeoffs: Achieving balance between broad sector classifications and specific product categories is difficult.
3. Data Quality: Precise carbon tracking is challenging and requires bith expertise in domain and granular supply of data.

Flamingo Model

The Flamingo Model is developed to address the challenge of incomplete Emission Impact Factor (EIF) datasets. It has the following challenges:

1. Incomplete EIF databases: These databases are often incomplete. It is particularly for novel materials or complex products.
2. Precision vs Recall Trade-offs: Flamingo exploits the hierarchical organisations of industry codes to navigate this trade-off.
3. Data Quality: Precise carbon tracking is challenging and requires bith expertise in domain and granular supply of data.

Future Trajectory/Development of the models

Parakeet Model

Future development of parakeet may include:

1. Multimodal Capabilities: Able to process visuals beyond text for better understanding of specific activities.
2. Dynamic Database Integration: Real-time updates related to emission factors.
3. Quantifying Uncertainity: Providing confidence levels for recommendations to improve decision-making.
4. Domain-wise Sector Learning: Getting insights from various domains to improve recommendations for novel activities.

CaML Model

Future development of the CaML model includes:

1. Semantic Understanding Refined: Enhanced model capability to simplify complex product descriptions.
2. Hierarchical Classification: implementation of multi-level sector mapping for products that covers multiple categories.
3. Integration with pLCA: Creating various hybrid approaches that combine the efficiency of EIO-LCA with the precision of process-based assessments.

Flamingo Model

Future development of the Flamingo Model includes:

1. Multimodal Capabilities: Able to process visuals beyond text for better understanding of specific activities.
2. Zero-Shot Learning: It can help the model predict industry codes for products new to the market without needing lots of training data.
3. Combining with other models: Combining it with Parakeet or CaML can create full systems for tracking carbon footprints.

CodeCarbon

Introduction

Code Carbon is an open source tool for tracking and reducing carbon emissions in computing. It is designed to estimate and track the carbon dioxide emissions produced by computing processes, with a particular focus on artificial intelligence and machine learning competitions. This tool is a significant step forward in the goal of making computing more sustainable and provides developers with the ability to measure, understand, and ultimately reduce the carbon footprint of the code execution.

CodeCarbon is developed by Mila, BCG GAMMA, Haverford College in Pennsylvania, and Comet.ml. The code carbon project aims to inspire the AI community to calculate, disclose, and reduce its carbon footprint.

Functionality and Implementation

CodeCarbon functions as lightweight software that seamlessly integrates into Python code. The tool operates by tracking hardware, electricity, and power consumption of components and then applies the carbon intensity of the geographical region where the computing is being performed to calculate emissions.

The software supports both online and offline mode, which is flexible for various computing environments. When operating in online mode, the tool uses real-time data about regional energy mixes to provide more accurate emission estimates. The tool measures emissions in kilograms of CO2 equivalent, which is a standardised measure used to express the global warming potential of various greenhouse gases.

Implementation approaches

Developers can implement the tool in their workflow using different approaches. Some of them are:

1. The straightforward method is to run the code on command line, where users can create a configuration file and monitor emissions without modifying their code
2. A specific tracking tool can be integrated directly to the python code
3. When using the explicit object approach, developers instantiate an EmissionTracker object and use the Start() and Stop() methods to begin and end the emission tracking 1st specific section of codes.

Practical application

Organisations develop systems that can use codecarbon to gain insights into the environmental costs of their operations. Integrating this information to produce sustainability initiatives and establishing emission reduction targets. Codecarbon contributes to responsibility and AI development by encouraging consideration of broad impacts as AI systems are increasingly integrated into various sectors and industries.