DEUTSCHE BUNDESBANK
EUROSYSTEM

Beyond Dictionaries: Machine-Learned Normalization of Emission Units in Information Extraction

Technical Report 2026-04

Lisa Reichenbach[1]
Malte Schierholz[2,3]
Anna Steinberg Schulten[2,3]
Simon Oehler[1]

[1]Deutsche Bundesbank, Research Data and Service Centre
[2]Ludwig-Maximilians-Universität München (LMU Munich)
[3]Munich Center for Machine Learning (MCML)

Disclaimer:

The views expressed in this technical report are personal views of the authors and do not necessarily reflect the views of the Deutsche Bundesbank or the Eurosystem.

Research Data and Service Centre

Abstract

Sustainability reports denote greenhouse gas emissions with heterogeneous unit strings, such as abbreviations, inconsistent formatting, and scale markers. In emission extraction pipelines, these units must be normalized to a common target, e.g. *t CO2e*, to enable reliable comparison and aggregation of emissions across companies, reporting years, scopes, and regions. We propose a hybrid normalization approach: If a raw unit string is not found in a curated dictionary, a machine learning classifier maps noisy unit strings to normalized emission-unit categories that can then be converted to the target unit. The classifier is trained using character-level TF-IDF, lightweight string features, and LLM-assisted data augmentation. On random-splits of the augmented dataset, an ensemble model achieves approximately 95% accuracy. Additional evaluation on rare real-world unit strings indicates that supervised fallback classifiers can handle out-of-dictionary variants in climate-related information extraction.

Use of AI Assistants

AI-based writing assistants, including GPT-based tools, were used to support language refinement, wording suggestions, and minor editorial improvements during manuscript preparation. All AI-generated suggestions were manually reviewed, revised where necessary, and verified by the authors. The authors remain fully responsible for the content, analysis, results, and conclusions presented in this paper.

Contact

Deutsche Bundesbank, Research Data and Service Centre (RDSC)
Corresponding author: lisa.reichenbach@bundesbank.de
Homepage: https://www.bundesbank.de/en/statistics/sustainability/greenhouse-gas-insights-and-sustainability-tracking-gist--992500

Keywords: unit normalization, information extraction, greenhouse gas emissions, sustainability reporting, supervised classification, data augmentation, emission metrics harmonization

Citation: Reichenbach, L., Schierholz, M., Steinberg Schulten, A., and Oehler, S. (2026). Beyond Dictionaries: Machine-Learned Normalization of Emission Units in Information Extraction, Technical Report 2026-04. Deutsche Bundesbank, Research Data and Service Centre

Contents

1 Introduction **4**

2 Related Work **5**

3 Methodology **6**
3.1 Hybrid Harmonization Pipeline 6
3.2 Task Formulation 6
3.3 Data and Feature Representation 6
3.4 Experimental Setup 7

4 Results **9**
4.1 Evaluation of Fallback Classifiers 9
4.2 Per-Class Performance and Error Analysis 9
4.3 Robustness on the Rare-Unit Dataset 11

5 Conclusion **12**

6 Limitations **13**

References **14**

A Appendix **16**
A.1 Prompt Template and Examples for LLM-assisted Data Augmentation 16
A.2 Class Distribution by Data Source 17
A.3 Implementation and Hyperparameter Details 17
A.4 Per-Class Classification Results 19
A.5 Detailed Results on the Rare-Unit Dataset 19

1 Introduction

Corporate sustainability disclosures are key sources for assessing greenhouse-gas emissions, climate-related risks, and decarbonization progress. As companies disclose growing volumes of Scope 1, Scope 2, and Scope 3 emissions under frameworks such as the GHG Protocol, TCFD, and CDP ([1], [2], [3]), traditional NLP- and LLM-based systems are increasingly used to extract structured climate-related information from these reports ([4], [5], [6]).

A persistent challenge in downstream analyses such as benchmarking, time-series comparisons, and total-emission calculations is that extracted emission-unit strings vary in language, capitalization, punctuation, abbreviation, and scale notation. As Table 1 illustrates, superficially similar strings may imply conversion factors that differ by several orders of magnitude. Extracted quantities must therefore be normalized to a common target unit, typically tonnes of CO2-equivalent (*t CO2e*).

Raw unit string	Norm. class	Factor
"Tonnes CO2e"	*t CO2e*	1
"mt CO2e"	*t CO2e*	1
"kg CO2 equivalent"	*kg CO2e*	0.001
"kilotonnes CO2e"	*kt CO2e*	1,000
"1,000 t CO2"	*kt CO2e*	1,000
"million t CO2e"	*Mt CO2e*	1,000,000
"US tons CO2e"	*USt CO2e*	0.907185
"kg CO2/kWh"	*Other*	–

Table 1: Examples of raw unit strings, normalized classes, and deterministic conversion factors to *t CO2e*.

Emission-unit normalization can be viewed as a post-processing step after value-unit pairs have been extracted by an upstream RAG-based pipeline ([7], [8]). Dictionary-based normalization handles known variants precisely, but is brittle to unseen spellings, multilingual expressions, abbreviations, and extraction artifacts, reflecting known limitations of rule-based and static-dictionary normalization approaches ([9]).

In our extraction setting, this brittleness is substantial: in a prior run over 1,760 corporate reports, exact dictionary lookup left 23.8% of extracted value-unit pairs unresolved. Moreover, 18.5% of all pairs were dictionary misses that could be manually assigned to specific emission-unit classes other than *Other*. This indicates that many unresolved strings are potentially convertible emission units and motivates a supervised fallback component for out-of-dictionary variants.

Yet unit harmonization is rarely treated as a standalone task in ClimateNLP, where prior work has focused mainly on document classification, climate-risk identification, metric extraction, and named entity recognition ([4], [6], [10], [11]).

We address this gap by formulating emission-unit normalization as a supervised text classification task within a hybrid dictionary-classifier pipeline. Dictionary lookup identifies known units, while a lightweight fallback classifier normalizes unknown unit strings. Using a manually curated dictionary augmented with LLM-generated variants ([12], [13]), we evaluate the approach across complementary evaluation settings covering both augmented and rare real-world unit strings.

2 Related Work

Normalization in information extraction is commonly treated as a post-processing step that maps heterogeneous surface forms to canonical representations. Unit and quantity normalization often relies on curated inventories, rule-based matching, and deterministic conversion tables, which work well for known surface forms but are less robust to noisy, abbreviated, or domain-specific variants ([9]). Early work on text and temporal normalization illustrates this pattern: CTEMP ([14]), for example, extracts and normalizes Chinese temporal expressions using a grammar- and constraint-rule-based chart parser. Related value-unit extraction has been studied in biomedical text, including the extraction of laboratory test names, numerical results, units, and contextual attributes from free text ([15]). More recent normalization and quantity-extraction systems combine lexical, statistical, or neural components with unit lookup and conversion rules ([19]). However, unit normalization is usually treated as auxiliary, with limited evaluation of fallback mechanisms for out-of-dictionary unit strings.

Climate-related information extraction has primarily addressed climate-document classification, risk identification, metric extraction, and named entity recognition in corporate disclosures ([4], [6], [10], [11]). Emission-unit harmonization is often handled implicitly through static dictionaries or downstream conversion rules, despite the fact that rare or noisy unit strings can lead to large aggregation errors. We address this gap by evaluating a hybrid dictionary-classifier approach for normalizing greenhouse-gas emission units in sustainability-report extraction.

3 Methodology

3.1 Hybrid Harmonization Pipeline

Emission extraction pipelines commonly require a unit-harmonization step after raw value-unit pairs have been extracted. In a deterministic dictionary-based approach, extracted raw unit strings are matched exactly against a curated dictionary. If no match is found, the unit remains unresolved and cannot be reliably converted to a common target unit.

We propose and evaluate a hybrid extension in which unresolved raw unit strings are passed to a supervised machine-learning fallback classifier, as shown in Figure 1. The fallback is designed to recognize unseen lexical variants of known unit classes, while unit categories outside the predefined label space still require dictionary extension or model retraining. In this work, we evaluate the fallback mechanism offline as a standalone unit-normalization component.

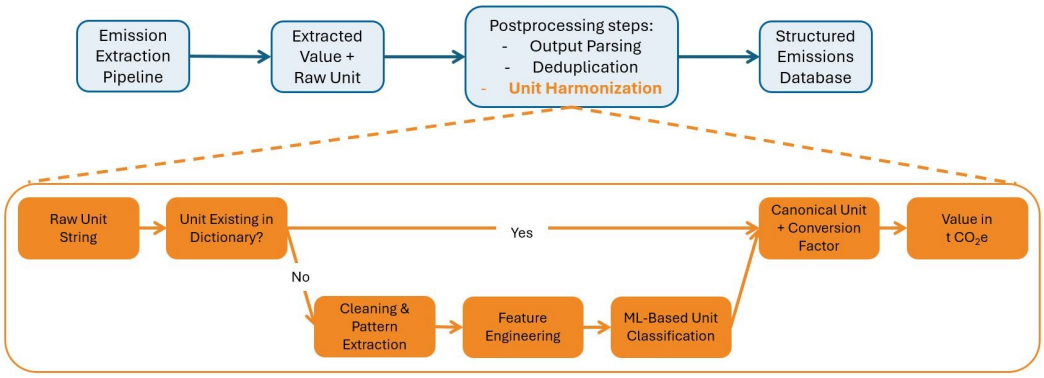


Figure 1: Overview of an emission extraction pipeline and the proposed hybrid extension for unit harmonization. In the evaluated extension, unresolved raw unit strings are passed to a machine-learning fallback classifier before the extracted values are converted to t CO_2e and stored in a structured emissions database.

3.2 Task Formulation

We formulate the fallback unit-harmonization component as a supervised classification task. Given an extracted raw unit string u that is not resolved by exact dictionary lookup, such as "million metric tonnes of CO2e", the model predicts a normalized unit class $c \in C$ such as *Mt CO2e* (see Table A2 for the complete class list). Each normalized emission-unit class, except non-convertible catch-all labels such as *Other*, is associated with a deterministic conversion factor f_c such that 1 unit of the normalized class = $f_c \times$ t CO_2e. For example, *Mt CO2e* corresponds to f_c = 1,000,000.

3.3 Data and Feature Representation

The dataset is based on a manually curated unit dictionary in which each instance corresponds to one distinct raw unit-string entry paired with its normalized unit class and conversion factor. The dictionary also stores occurrence frequencies from previous applications of the emission-extraction pipeline to sustainability-report collections. These frequencies indicate how often a raw string was

extracted across reports, reporting years, and scopes, but they do not create duplicate dataset instances and are used only for analysis, not as predictors. All frequency metadata were collected before the machine-learning experiments.

Given the highly imbalanced classes, as evident from the column "Manual" in Table A2, we augment underrepresented classes with GPT-4.1-generated variants ([20]), following prior LLM-based augmentation work under data scarcity and class imbalance ([12], [13], [21]). The augmentation is designed to expand existing semantic classes with plausible alternative surface forms, rather than to introduce new unit concepts. Because small changes in prefixes, denominators, or scale expressions can alter unit semantics, all generated candidates are deduplicated against the manual dictionary and manually validated. Lexical overlap among variants should be considered when interpreting performance estimates based on random data splits. The augmentation prompt is shown in Figure A1, and randomly selected examples are provided in Table A1 in Appendix A.1. The final dataset comprises 1,849 unit strings across eight target classes, including 471 manually validated LLM-generated variants. Table A2 reports the resulting class distribution.

The classifiers use only features derived from the raw unit string. Unit strings are lowercased, normalized, and represented using character-level TF-IDF features. Lowercasing was retained because it improved validation performance in preliminary experiments, despite discarding potentially informative case distinctions such as *mt* versus *Mt*; we revisit this trade-off in the error analysis and limitations. Additional regex-based indicators capture scale and quantity expressions such as *kilo*, *million*, *mega*, or *short tons*; simple numerical features such as string length are also included. Textual features are vectorized using character n-grams and combined with scaled numerical features in a *ColumnTransformer*. Target labels are label-encoded during training and decoded after prediction.

Full implementation details, including preprocessing steps, character n-gram settings, hyperparameter grids, selected model configurations, software versions, and random seeds, are provided in Appendix A.3. The code and unit dictionary are available in a public GitHub repository[1)] to support reproducibility and future work.

3.4 Experimental Setup

As a trivial lower bound, we include a majority-class classifier that always predicts the most frequent normalized unit class in the training data, namely *Other*. Since exact dictionary lookup on the curated dataset is perfect by construction, we do not report it. We compare the majority-class classifier with several supervised models for the fallback task: Logistic Regression, Random Forest, Support Vector Machine (SVM), XGBoost, LightGBM, Multi-layer Perceptron (MLP), and a majority-voting ensemble classifier combining SVM, MLP, and Logistic Regression. Hyperparameters are optimized on the full dataset using grid search with stratified cross-validation.

Evaluation is conducted in two complementary settings. The **random-split dataset** evaluates model performance under a standard stratified split of the full dataset, including both manually curated and LLM-augmented unit strings. The **rare-unit dataset** is a robustness test split comprising manually curated unit strings whose exact raw surface forms occurred at most five times in the

1 https://github.com/gist-sustainability/unit_harmonization/releases/tag/v0.1.0

prior extraction runs, based on the frequency metadata defined in Section 3.3. The rare-unit test set was constructed from pre-existing dictionary entries and occurrence metadata, independently of model predictions and error analyses. LLM-generated variants are excluded because they were not observed in the extraction output and therefore do not have occurrence counts.

In the rare-unit experiment, the rare-unit test strings are excluded from model training, while LLM-augmented entries remain available in the training data to improve coverage of underrepresented classes. No additional hyperparameter tuning is performed. We instead use the hyperparameters selected in the preceding grid-search cross-validation experiments. Since that cross-validation used random splits of the full augmented dataset, some rare-unit strings may have contributed to hyperparameter selection. Thus, the rare-unit evaluation uses a fixed configuration and excludes the rare test strings from model fitting, but it is not fully independent of the preceding tuning procedure. Its purpose is to assess robustness to naturally observed low-frequency unit variants in the fallback use case.

We report accuracy, weighted F1-score ($F1_W$), and macro F1-score. In this single-label multi-class setting, accuracy is equivalent to micro-F1, so micro-F1 is not reported separately. Weighted F1 is used as the main imbalance-aware F1 metric because it accounts for class support and reflects the influence of frequent classes such as *Other*, *t CO2e*, and *kt CO2e*. Because aggregate metrics do not capture the practical severity of different error types, we complement them with per-class results and row-normalized confusion matrices, focusing on whether errors involve distant unit scales, such as *t CO2e* versus *Mt CO2e*, or conservative predictions of *Other*.

4 Results

4.1 Evaluation of Fallback Classifiers

Table 2 summarizes the offline performance of the fallback classifiers. We report 10-fold stratified cross-validation on the full augmented dictionary-derived dataset using *StratifiedKFold* with shuffling, fitting the complete preprocessing and classification pipeline separately within each fold. In addition, we evaluate each model in a single stratified 80/20 random-split run. Hyperparameters are fixed to the selected configurations reported in Table A4.

All supervised models clearly outperform the majority-class lower bound, whose weighted F1-score remains below 0.15. The Voting Ensemble achieves the strongest cross-validation performance across all metrics, with a weighted F1-score of 0.9552 ± 0.0113. Its comparatively low standard deviation also indicates stable performance across folds. On the random-split dataset, the ensemble remains among the best-performing models, with only MLP achieving higher performance in the single split run. Based on its consistently strongest cross-validation performance and competitive random-split results, we choose the Voting Ensemble as the selected fallback classifier.

The selected ensemble combines SVM, MLP, and Logistic Regression. It was chosen as a compact performance-runtime trade-off, since larger ensembles yielded only marginally higher validation scores while increasing runtime and complexity.

Model	10-fold CV			Random-split run		
	Acc.	$F1_W$	Macro F1	Acc.	$F1_W$	Macro F1
Majority-class	0.3067 ± 0.0026	0.1439 ± 0.0022	0.0587 ± 0.0004	0.3054	0.1429	0.0585
Logistic Regression	0.9524 ± 0.0026	0.9525 ± 0.0143	0.9621 ± 0.0103	0.9595	0.9596	0.9669
Random Forest	0.9421 ± 0.0189	0.9421 ± 0.0189	0.9551 ± 0.0134	0.9351	0.9355	0.9495
SVM	0.9516 ± 0.0116	0.9414 ± 0.0116	0.9542 ± 0.0068	0.9486	0.9489	0.9601
XGBoost	0.9448 ± 0.0149	0.9447 ± 0.0150	0.9570 ± 0.0107	0.9514	0.9515	0.9618
LightGBM	0.9486 ± 0.0165	0.9486 ± 0.0164	0.9598 ± 0.0144	0.9514	0.9515	0.9625
MLP	0.9492 ± 0.0103	0.9493 ± 0.0103	0.9603 ± 0.0081	**0.9649**	**0.9648**	**0.9751**
Voting Ensemble	$\mathbf{0.9551 \pm 0.0114}$	$\mathbf{0.9552 \pm 0.0113}$	$\mathbf{0.9671 \pm 0.0082}$	0.9595	0.9595	0.9708

Table 2: Offline evaluation of fallback classifiers for unit harmonization. Cross-validation results report mean $\pm$ standard deviation across 10 stratified folds on the full augmented dictionary-derived dataset. Random-split results are computed from a single stratified 80/20 train-test split on the random-split dataset.

4.2 Per-Class Performance and Error Analysis

To interpret the aggregate results, we examine the per-class performance of the selected fallback classifier, the Voting Ensemble, on the test portion of the random-split dataset. Figure 2 shows the row-normalized confusion matrix for the emission unit categories with a strong diagonal, indicating high recall across classes.

Most remaining errors involve related unit categories or predictions of *Other*. This distinction is important because the practical severity of misclassifications is not uniform: predicting *Other* for a convertible unit mainly leads to a missed automatic conversion, whereas confusing distant scale classes, such as *t CO2e* and *Mt CO2e*, would directly induce errors of several orders of magnitude.

The confusion matrix shows that such severe scale-related confusions are rare in the random-split dataset. The per-class scores in Appendix A.4 confirm this pattern, with all F1-scores above 0.92. This suggests that the fallback classifier is generally reliable for distinguishing emission unit categories.



Figure 2: Confusion matrix of emission unit categories for the random-split dataset. Each row is normalized to 100%, showing the distribution of predicted unit categories for each true unit category. The diagonal entries correspond to the recall of the respective categories.

The remaining errors in the random-split dataset are concentrated in linguistically ambiguous or under-specified cases rather than in systematic confusions between distant unit scales. For example, capitalization-sensitive abbreviations can lead to errors: lowercase *mt* denotes metric tonnes and corresponds to *t CO2e*, whereas uppercase *Mt* denotes megatonnes and corresponds to *Mt CO2e*. As shown in Table 3, other errors involve abbreviated scale markers such as *th.*, units embedded in context, missing mass units, or boundary cases such as emission intensities.

Raw string	True	Pred.	Error type
mt CO_2e	*t CO2e*	*Mt CO2e*	case-sensitive scale ambiguity
metric tons CO2 equivalent (based on 2016 data)	*t CO2e*	*Other*	unit embedded in context
emissions	*Other*	*t CO2e*	no mass unit
*kgCO2e/sqm*year*	*Other*	*kg CO2e*	intensity instead of absolute unit
th. t CO2 Eq.	*kt CO2e*	*t CO2e*	abbreviated scale marker

Table 3: Representative misclassifications of the selected fallback classifier on the random-split dataset.

4.3 Robustness on the Rare-Unit Dataset

As described in Section 3.4, the rare-unit test set consists of distinct manually curated real-world unit-string entries whose raw surface forms occurred at most five times in prior extraction runs. The resulting split is more imbalanced than the random-split dataset; the full train-test composition is reported in Appendix A.5, Table A6.

In this setting, the Voting Ensemble achieves an accuracy of 94.85%, a weighted F1-score of 94.84%, and a macro F1-score of 96.93%. The complete results are shown in Appendix Table A7. The weighted F1-score differs from the corresponding random-split result by only 1.11 percentage points, indicating strong performance on rare, naturally occurring unit variants rather than performance driven solely by frequent or LLM-augmented strings. As noted in Section 3.4, however, this evaluation is not fully independent of the preceding hyperparameter selection.

Detailed model comparisons, per-class scores, and the corresponding confusion matrix are provided in Appendix A.5. The rare-unit confusion matrix shows an even more conservative error pattern than the random-split evaluation: almost all remaining misclassifications are predictions of *Other*, while direct confusions between distant conversion-relevant unit scales are nearly absent.

5 Conclusion

This paper studies emission-unit normalization as a post-processing step for greenhouse gas emission extraction. We propose a hybrid approach that keeps deterministic dictionary lookup as a high-confidence first step and adds a supervised fallback classifier for out-of-dictionary unit strings. Using a curated and LLM-augmented dataset, we train different lightweight classifiers on character-level TF-IDF representations and simple domain-specific string features.

The results show that supervised classifiers can effectively extend dictionary coverage to unseen variants of known emission-unit classes, including rare real-world strings. The best-performing models achieve around 95% accuracy and weighted F1-score on both the random-split dataset and the rare-unit dataset. This suggests that the approach is not limited to frequent or augmented examples and can also handle rare real-world unit strings. Error analysis further suggests that most remaining mistakes stem from genuinely ambiguous abbreviations, missing unit information, or boundary cases such as emission intensities, rather than systematic large-scale conversion errors. The hybrid setup therefore provides a transparent and practical path toward more robust unit harmonization in climate-related information extraction.

6 Limitations

This study evaluates unit normalization as a standalone offline component rather than as part of an end-to-end emission extraction pipeline. Consequently, the evaluation does not capture interactions with upstream extraction errors, document context, table structure, or downstream aggregation. In practice, such errors may propagate, and scale confusions between units such as *t CO2e*, *kt CO2e*, and *Mt CO2e* can lead to conversion errors of several orders of magnitude.

A further limitation is the use of augmented dictionary-derived data. Although all LLM-generated variants were manually validated, random splits may overestimate generalization because highly similar surface forms can appear in both training and test data. The rare-unit dataset mitigates this by excluding LLM-augmented entries from the test set and evaluating only on rare manually curated real-world strings. However, it is not fully independent of model selection, as some rare-unit strings may have appeared in training or validation folds during the preceding random-split grid-search cross-validation. It may therefore be less conservative than a fully untouched test set. Moreover, the rare strings come from previous extraction runs and are not a prospective evaluation on newly extracted units from unseen reports. We do not include a mixed rare-plus-LLM holdout in order to keep the rare-unit test real-world-only, although such a synthetic holdout could be useful for sensitivity analysis on underrepresented classes.

Finally, the classifier is restricted to predefined unit classes and cannot handle genuinely new unit types without dictionary extension or retraining. Ambiguous abbreviations, multilingual expressions, and intensity-based units remain challenging. In particular, lowercasing improves robustness to inconsistent capitalization but removes informative distinctions such as *mt* versus *Mt*. Future work should integrate the classifier into the full extraction pipeline, incorporate document or table context, add case-sensitive or confidence-based rules for high-impact abbreviations, and use severity-aware metrics that weight errors by their implied conversion-factor difference.

References

[1] GHG Protocol, "The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard: Revised Edition," World Resources Institute; World Business Council for Sustainable Development, 2011. Available: https://ghgprotocol.org/sites/default/files/standards/ghg-protocol-revised.pdf

[2] TCFD, "Recommendations of the Task Force on Climate-related Financial Disclosures," Task Force on Climate-related Financial Disclosures, 2017. Available: https://www.fsb-tcfd.org/recommendations/

[3] CDP, "CDP Corporate Questionnaire 2024." https://www.cdp.net/en/disclose/question-bank, 2024.

[4] J. A. Bingler, M. Kraus, M. Leippold, and N. Webersinke, "Cheap talk and cherry-picking: What ClimateBert has to say on corporate climate risk disclosures," *Finance Research Letters*, vol. 47, p. 102776, 2022, doi: https://doi.org/10.1016/j.frl.2022.102776.

[5] E. A. L. Auzepy Alix AND Tönjes, "Evaluating TCFD reporting—A new application of zero-shot analysis to climate-related financial disclosures," *PLOS ONE*, vol. 18, no. 11, pp. 1–23, Nov. 2023, doi: 10.1371/journal.pone.0288052.

[6] A. Dave, M. Zhu, D. Hu, and S. Tiwari, "Climate AI for Corporate Decarbonization Metrics Extraction," arXiv.org, Papers 2411.03402, Nov. 2024. Available: https://ideas.repec.org/p/arx/papers/2411.03402.html

[7] A. Dimmelmeier *et al.*, "Informing climate risk analysis using textual information - A research agenda," in *Proceedings of the 1st workshop on natural language processing meets climate change (ClimateNLP 2024)*, D. Stammbach, J. Ni, T. Schimanski, K. Dutia, A. Singh, J. Bingler, C. Christiaen, N. Kushwaha, V. Muccione, S. A. Vaghefi, and M. Leippold, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 12–26. doi: 10.18653/v1/2024.climatenlp-1.2.

[8] Y. Zou *et al.*, "ESGReveal: An LLM-based approach for extracting structured data from ESG reports," *Journal of Cleaner Production*, vol. 489, p. 144572, 2025, doi: https://doi.org/10.1016/j.jclepro.2024.144572.

[9] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of Non-Standard Words," *Computer Speech & Language*, vol. 15, pp. 287–333, Jul. 2001, doi: 10.1006/csla.2001.0169.

[10] N. Webersinke, M. Kraus, J. A. Bingler, and M. Leippold, "ClimateBert: A pretrained language model for climate-related text," *CoRR*, vol. abs/2110.12010, 2021, Available: https://arxiv.org/abs/2110.12010

[11] H. Pan, M. Adamu, Q. Zhang, E. Dragut, and L. J. Latecki, "ClimateIE: A Dataset for Climate Science Information Extraction," in *Proceedings of the 2nd workshop on natural language processing meets climate change (ClimateNLP 2025)*, K. Dutia, P. Henderson, M. Leippold, C. Manning, G. Morio, V. Muccione, J. Ni, T. Schimanski, D. Stammbach, A. Singh, A. (Ruiran). Su, and S. A. Vaghefi, Eds., Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 76–98. doi: 10.18653/v1/2025.climatenlp-1.6.

[12] B. Ding *et al.*, "Data Augmentation using LLMs: Data Perspectives, Learning Paradigms and Challenges," in *Findings of the association for computational linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 1679–1705. doi: 10.18653/v1/2024.findings-acl.97.

[13] H. Dai *et al.*, "AugGPT: Leveraging ChatGPT for Text Data Augmentation," *IEEE Transactions on Big Data*, vol. 11, no. 3, pp. 907–918, 2025, doi: 10.1109/TBDATA.2025.3536934.

[14] W. Mingli, L. Wenjie, L. Qin, and L. Baoli, "CTEMP: A chinese temporal parser for extracting and normalizing temporal information," in *Natural language processing – IJCNLP 2005*, R. Dale, K.-F. Wong, J. Su, and O. Y. Kwong, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 694–706. Available: https://aclanthology.org/I05-1061.pdf

[15] Y. S. Kang and M. Kayaalp, "Extracting laboratory test information from biomedical text," *Journal of Pathology Informatics*, vol. 4, no. 1, p. 23, 2013, doi: https://doi.org/10.4103/2153-3539.117450.

[16] L. Lange, J. Strötgen, H. Adel, and D. Klakow, "Multilingual Normalization of Temporal Expressions with Masked Language Models," in *Proceedings of the 17th conference of the european chapter of the association for computational linguistics*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1174–1186. doi: 10.18653/v1/2023.eacl-main.84.

[17] M. Ashmawy, M. Fakhr, and F. Maghraby, "Lexical Normalization Using Generative Transformer Model (LN-GTM)," *International Journal of Computational Intelligence Systems*, vol. 16, Nov. 2023, doi: 10.1007/s44196-023-00366-8.

[18] L. Foppiano, L. Romary, M. Ishii, and M. Tanifuji, "Automatic Identification and Normalisation of Physical Measurements in Scientific Literature," in *Proceedings of the ACM symposium on document engineering 2019*, in DocEng '19. New York, NY, USA: Association for Computing Machinery, 2019. doi: 10.1145/3342558.3345411.

[19] S. Almasian, V. Kazakova, P. Göldner, and M. Gertz, "CQE: A Comprehensive Quantity Extractor," in *Proceedings of the 2023 conference on empirical methods in natural language processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 12845–12859. doi: 10.18653/v1/2023.emnlp-main.793.

[20] OpenAI, "Introducing GPT-4.1 Model Family." Apr. 2025. Available: https://openai.com/index/gpt-4-1/

[21] J. Ye *et al.*, "LLM-DA: Data Augmentation via Large Language Models for Few-Shot Named Entity Recognition," *ArXiv*, vol. abs/2402.14568, 2024, Available: https://api.semanticscholar.org/CorpusID:267782766

A Appendix

A.1 Prompt Template and Examples for LLM-assisted Data Augmentation

To mitigate class imbalance in the unit normalization data, we use an LLM to generate additional variants for underrepresented unit categories. For each target category *c*, the model is prompted with examples from the well-represented *t CO2e* class and instructed to produce realistic unit strings with analogous surface-form variation patterns, including differences in spelling, formatting, abbreviations, whitespace, special characters, and capitalization. The prompt also restricts the output to variants without additional parenthetical descriptions and requests a Python list for straightforward downstream processing.

The prompt template used for the generation is shown in Figure A1. Here, *c* denotes the target unit category, while {*tco2e_variants_list*} denotes the example list from the well-represented *t CO2e* class.

Augmentation prompt

Please generate a list of 100 unique variants for the unit category *c*, following the style and diversity of the example list below for "t CO2e". The variants should differ in spelling, formatting, whitespaces, abbreviations, use of special characters and/or capitalization.
Do not include any variants with additional descriptions in parentheses, e.g., "(per shipment)", "(biogenic)", etc.
Please avoid duplicates and ensure that all variants are realistic and could plausibly appear in real-world data. The output should be a Python list.
Here is my example list:
{*tco2e_variants_list*}

Figure A1: Prompt template used for LLM-assisted data augmentation. The placeholder *c* is replaced by the target unit category, while {*tco2e_variants_list*} is replaced by the example variants from the well-represented *t CO2e* class.

To illustrate the realized diversity of the augmentation, we manually inspected the LLM-generated variants from the final augmented dictionary. Table A1 presents randomly selected examples from different target classes and summarizes the main types of observed surface-form variation, including capitalization, hyphenation, abbreviations, lexical expansions, and Unicode characters. Although some variants remain close to existing dictionary entries, the examples demonstrate that the augmentation produces a range of plausible surface-form realizations within predefined semantic classes.

Target class	Generated variant	Observed variation
MM Mt CO2e	*mm mt CO2-Equivalent*	case, hyphen, expansion
kg CO2e	*kg CO2-Equiv*	hyphen, abbreviation
MM Mt CO2e	*mm mt CO_2e*	case, Unicode subscript
MM Mt CO2e	*MMMt CO_2-equivalent*	abbreviation, Unicode, hyphen
10 kt CO2e	*10 KT CO2 equivalent*	case, expansion
USt CO2e	*us ton CO_2 eq*	case, singular, abbreviation
kg CO2e	*Kg CO2-Eq*	case, hyphen, abbreviation
10 kt CO2e	*10 kt CO2-Äq*	hyphen, German abbreviation

Table A1: Randomly selected examples of manually validated LLM-generated unit-string variants. The final column summarizes the main types of observed surface-form variation.

A.2 Class Distribution by Data Source

Table A2 reports the class distribution of the final unit-string dataset across all eight target classes, distinguishing between manually curated entries and validated LLM-generated variants. The augmentation was applied selectively to classes with limited manual coverage, while already well-represented classes were kept unchanged. The category *Other* contains strings that cannot be unambiguously assigned to a target emission-unit class or refer to incompatible quantities, such as "emissions", "kWh", or "t CO2e/million euros".

Unit class	Manual (Proportion)	Augmented	Total
t CO2e	343 (24.9%)	0	343
kt CO2e	249 (18.1%)	0	249
Mt CO2e	184 (13.4%)	0	184
kg CO2e	25 (1.8%)	101	126
USt CO2e	4 (0.3%)	112	116
10 kt CO2e	4 (0.3%)	116	120
MM Mt CO2e	2 (0.1%)	142	144
Other	567 (41.1%)	0	567
Total	**1378**	**471**	**1849**

Table A2: Class distribution of unit strings by data source. Manual entries originate from the curated unit dictionary, whereas augmented entries are validated LLM-generated variants.

A.3 Implementation and Hyperparameter Details

All models were implemented in Python version 3.11.9 using scikit-learn version 1.9.1, with XGBoost version 3.2.0 and LightGBM version 4.7.0. Unless stated otherwise, we used the default parameter settings of the respective libraries. Random seeds were fixed to 42 for data splitting and model training where applicable.

Hyperparameters were selected on the training portion using grid search with 5-fold stratified cross-validation and weighted F1-score as the refit criterion. The selected pipeline was then refitted on the full training portion and evaluated on the held-out test set. Raw unit strings were represented by a concatenated text feature and one numerical length feature. The text feature combined: (i) a lowercased raw string in which numeric substrings were replaced by *NUMBER*; (ii) a regex-based quantity keyword covering markers such as *kilo*, *thousand*, *kt*, *million*, *mio.*, *mega*, *kg* and *US/short tons*; and (iii) a cleaned unit-base string obtained by removing numeric expressions and non-word characters, removing the token *of*, stripping whitespace, and lowercasing. We also tested case-preserving variants of components (i) and (iii), as well as variants preserving case in only one of the two components. These alternatives performed worse in preliminary validation experiments and were therefore not used in the final configuration.

The concatenated text field was vectorized using character-level TF-IDF with *analyzer=char_wb*, *ngram_range=(1,3)*, *min_df=3*, *max_df=0.95*, *max_features=1000*, and *sublinear_tf=True*; all other vectorizer parameters were left at their scikit-learn defaults, including L2 normalization. The numerical length feature was standardized using *StandardScaler*.

The complete hyperparameter search spaces are summarized in Table A3. For the Voting En-

semble, we first evaluated all base-model subsets of size three to six and then selected the final subset based on the trade-off between predictive performance, runtime, and model complexity. Table A4 reports the final model configurations used for the classifier comparison in the main experiments.

Model	Hyperparameter grid
Logistic Regression	Two grids were evaluated. First, $C \in \{0.01, 0.1, 1.0, 10.0, 100.0, 1000.0\}$, *penalty* $\in \{l2\}$, *class_weight* $\in \{None, balanced\}$, *solver* $\in \{lbfgs, newton\text{-}cg\}$, *max_iter* $\in \{1000, 2000, 3000\}$. Second, $C \in \{0.01, 0.1, 1.0, 10.0, 100.0\}$, *penalty* $\in \{l1, l2\}$, *class_weight* $\in \{None, balanced\}$, *solver* $\in \{saga\}$, *max_iter* $\in \{2000\}$
SVM	$C \in \{0.1, 1, 10, 100\}$, *kernel* $\in \{linear, rbf\}$, *gamma* $\in \{scale, auto\}$, *class_weight* $\in \{None, balanced\}$
Random Forest	*n_estimators* $\in \{100, 200, 500\}$, *max_depth* $\in \{None, 10, 20\}$, *min_samples_split* $\in \{2, 10\}$, *min_samples_leaf* $\in \{1, 4\}$, *max_features* $\in \{sqrt, log2\}$, *bootstrap* $\in \{True, False\}$, *class_weight* $\in \{None, balanced\}$
XGBoost	*max_depth* $\in \{4, 6\}$, *learning_rate* $\in \{0.2\}$, *n_estimators* $\in \{100, 300\}$, *subsample* $\in \{1\}$, *colsample_bytree* $\in \{0.8\}$, *gamma* $\in \{None, 0.3\}$, *reg_alpha* $\in \{None, 0.1\}$, *reg_lambda* $\in \{None, 1.5\}$
LightGBM	*num_leaves* $\in \{31, 63\}$, *max_depth* $\in \{-1, 6, 8\}$, *n_estimators* $\in \{100, 200, 600\}$, *learning_rate* $\in \{0.05, 0.1\}$, *feature_fraction* $\in \{0.8, 1.0\}$, *bagging_freq* $\in \{0, 1\}$, *reg_lambda* $\in \{0.0, 2.0\}$
MLP	*hidden_layer_sizes* $\in \{(100,), (200,), (100, 100), (200, 100)\}$, *activation* $\in \{relu, tanh\}$, *alpha* $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$, *learning_rate_init* $\in \{0.001, 0.01\}$, *solver* $\in \{adam, lbfgs\}$, *max_iter* $\in \{500, 1000\}$, *early_stopping* $\in \{True, False\}$
Voting Ensemble	All base-model subsets of size 3 to 6 were evaluated from {XGBoost, LightGBM, Logistic Regression, Random Forest, SVM, MLP}, yielding 42 model combinations. The final subset, consisting of SVM, MLP, and Logistic Regression, was selected based on the trade-off between performance, runtime, and model complexity, as larger ensembles provided only marginal performance gains. For this subset, *voting* $\in \{hard, soft\}$ was tested; *weights=None*.

Table A3: Hyperparameter grids used for grid search cross-validation.

Model	Selected hyperparameters
Logistic Regression	*C=100*, *penalty=l2*, *class_weight=balanced*, *solver=lbfgs*, *max_iter=1000*
SVM	*C=100*, *kernel=linear*, *gamma=scale*, *class_weight=None*, *probability=True*, *random_state=42*
Random Forest	*n_estimators=200*, *max_depth=None*, *bootstrap=False*, *class_weight=None*, *max_features=sqrt*, *min_samples_leaf=1*, *min_samples_split=2*, *random_state=42*, *n_jobs=-1*
XGBoost	*n_estimators=100*, *max_depth=6*, *learning_rate=0.2*, *subsample=1.0*, *colsample_bytree=0.8*, *gamma=None*, *reg_alpha=None*, *reg_lambda=1.5*, *random_state=42*, *eval_metric=mlogloss*
LightGBM	*n_estimators=600*, *max_depth=6*, *learning_rate=0.1*, *bagging_freq=0*, *feature_fraction=1.0*, *num_leaves=31*, *reg_lambda=0.0*, *random_state=42*, *objective=multiclass*, *metric=multi_logloss*, *verbosity=-1*
MLP	*hidden_layer_sizes=(200, 100)*, *activation=tanh*, *solver=lbfgs*, *alpha=0.001*, *learning_rate_init=0.001*, *max_iter=500*, *random_state=42*, *early_stopping=True*
Voting Ensemble	*voting=soft*, *weights=None*; base models: SVM, MLP, Logistic Regression

Table A4: Selected hyperparameters used for the classifier comparison.

A.4 Per-Class Classification Results

Table A5 reports the per-class precision, recall, and F1-scores of the selected fallback classifier on the test portion of the random-split dataset. The results complement the aggregate metrics and the confusion matrix in Section 4.2 by showing that performance is high across the eight unit categories. Most classes achieve F1-scores close to or above 0.95, with perfect scores for *USt CO2e* and *MM Mt CO2e*. Class *t CO2e* exhibits the lowest F1-score, driven by moderately lower precision and recall compared to the other classes. *kt CO2e* shows the lowest recall. The *Other* class remains robust, suggesting that incompatible or non-convertible units are usually not forced into a specific emission-unit category.

Class	Precision	Recall	F1	Support
t CO2e	0.9275	0.9275	0.9275	69
kt CO2e	0.9792	0.9400	0.9592	50
Mt CO2e	0.9024	1.0000	0.9487	37
kg CO2e	0.9615	1.0000	0.9804	25
USt CO2e	1.0000	1.0000	1.0000	23
10 kt CO2e	1.0000	1.0000	1.0000	24
MM Mt CO2e	1.0000	1.0000	1.0000	29
Other	0.9636	0.9381	0.9507	113

Table A5: Per-class performance of the selected classifier on the test portion of the random-split dataset. Support denotes the number of test instances per class.

A.5 Detailed Results on the Rare-Unit Dataset

The rare-unit evaluation uses a fixed train-test split designed to assess robustness on naturally occurring low-frequency unit strings. The test set consists only of distinct manually curated real-world unit-string entries whose corresponding raw surface forms occurred at most five times in prior extraction runs. These rare-test strings were excluded from training. The training set contains the remaining manually curated entries and all validated LLM-generated variants. LLM-generated variants are not included in the rare-unit test set because this evaluation is intended to focus on naturally observed rare strings. Table A6 summarizes the resulting train-test composition by class.

Class	Manual total	Rare test	Manual train	Aug. train	Total train
t CO2e	343	68	275	0	275
kt CO2e	249	49	200	0	200
Mt CO2e	184	36	148	0	148
kg CO2e	25	19	6	101	107
USt CO2e	4	4	0	112	112
10 kt CO2e	4	1	3	116	119
MM Mt CO2e	2	1	1	142	143
Other	567	113	454	0	454
Total	1378	291	1087	471	1558

Table A6: Train-test composition of the rare-unit evaluation setting. The rare test set contains only manually curated real-world unit strings whose exact raw surface forms occurred at most five times in prior extraction runs. LLM-generated variants are excluded from the test set and used only for training.

Table A7 reports the performance of all candidate fallback classifiers on this rare-unit dataset. The split composition in Table A6 shows that the rare-unit test set is substantially more imbalanced than the random-split dataset, with very small support for *USt CO2e*, *10 kt CO2e*, and *MM Mt CO2e*. The aggregate metrics should therefore be interpreted together with the per-class results below. The Voting Ensemble achieves the strongest results on this dataset and shows the smallest absolute difference in weighted F1-score relative to the random-split dataset. This is particularly relevant because the MLP achieved the highest score on the random-split dataset, but exhibits a larger drop on the rare-unit dataset. The results therefore support selecting the Voting Ensemble as the more robust fallback classifier.

Model	Acc.	$F1_w$	Macro F1	Δ $F1_w$
Log. Regression	0.9416	0.9413	0.9620	0.0183
Random Forest	0.9141	0.9125	0.8228	0.0230
SVM	0.9175	0.9174	0.8260	0.0315
XGBoost	0.8900	0.8892	0.7701	0.0623
LightGBM	0.9038	0.9035	0.8600	0.0480
MLP	0.9347	0.9345	0.9593	0.0303
Voting Ensemble	**0.9485**	**0.9484**	**0.9693**	**0.0111**

Table A7: Robustness evaluation of the fallback classifiers on the rare-unit dataset. Δ $F1_w$ denotes the absolute difference in weighted F1-score between the rare-unit dataset and the test portion of the random-split dataset.

Table A8 provides the corresponding per-class results for the Voting Ensemble on the rare-unit dataset. Performance remains high across the main classes, with F1-scores above 0.91 for *t CO2e*, *kt CO2e*, *Mt CO2e*, *kg CO2e*, and *Other*. The class *Other* also remains reliable, with a recall of 0.9735, suggesting that rare non-target or incompatible unit strings are usually not forced into one of the emission-unit classes. For classes with very small rare-test support, especially *USt CO2e*, *10 kt CO2e*, and *MM Mt CO2e*, the perfect scores should not be overinterpreted. Overall, the per-class results indicate that the classifier generalizes well to rare real-world unit variants and that the strong aggregate performance is not solely explained by frequent classes or by evaluation on LLM-augmented strings.

Class	Precision	Recall	F1	Support
t CO2e	0.9254	0.9118	0.9185	68
kt CO2e	0.9400	0.9592	0.9495	49
Mt CO2e	1.0000	0.9167	0.9565	36
kg CO2e	1.0000	0.9474	0.9730	19
USt CO2e	1.0000	1.0000	1.0000	4
10 kt CO2e	1.0000	1.0000	1.0000	1
MM Mt CO2e	1.0000	1.0000	1.0000	1
Other	0.9402	0.9735	0.9565	113

Table A8: Per-class performance of the selected fallback classifier on the rare-unit dataset. Support denotes the number of test instances per class.

Figure A2 shows the row-normalized confusion matrix for the rare-unit dataset, analogously to Figure 2. Compared to the random-split dataset, the error pattern is more conservative: a lot of the remaining misclassifications are predictions of *Other*, rather than confusions between alternative specific unit scales. This is relevant because errors are not equally severe in downstream emission aggregation. A false *Other* prediction prevents automatic conversion and may require manual review, whereas confusing units with very different conversion factors, such as *kg CO2e* and *Mt CO2e*, would directly distort normalized emissions by several orders of magnitude. For example,

the errors for *kg CO2e* are entirely assigned to *Other*. Direct confusions between specific emission units are rare and occur only in small proportions. This suggests that the classifier tends to avoid assigning a potentially wrong conversion-relevant unit category when uncertain, reducing the risk of severe scale-related conversion errors.

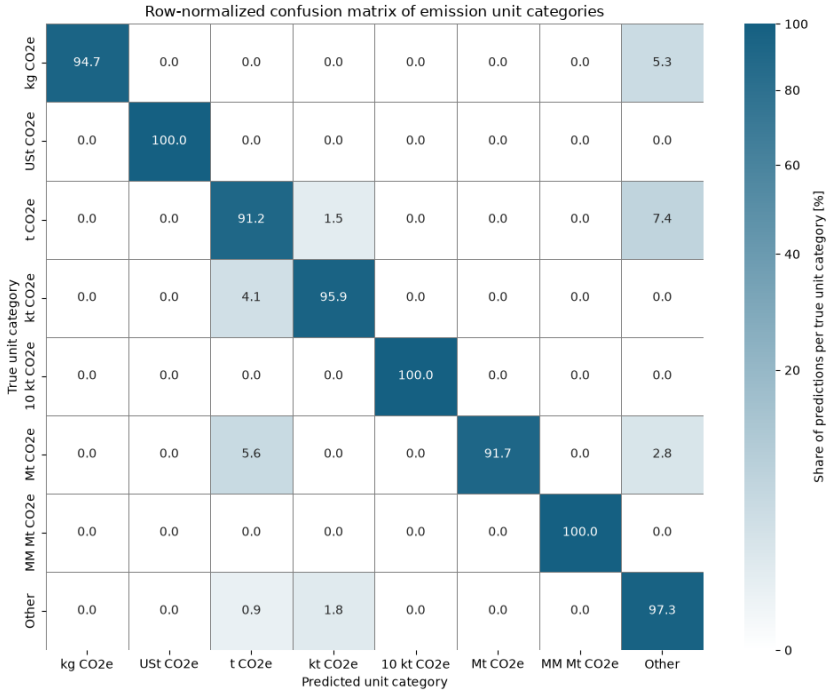


Figure A2: Row-normalized confusion matrix of predicted emission-unit categories on the rare-unit dataset. Values are percentages per true category; diagonal entries correspond to category-wise recall.