CanWeTrustReFAIR: A Replication Study

Ahmed Y. Radwan[1], Claudia Farkas[1], and Amir Haeri[1]
[1]Department of Electrical Engineering and Computer Science, York University, Toronto, ON M3J 1P3, Canada

***Abstract*—ReFAIR proposes a context-aware requirements engineering framework to support fairness in machine learning (ML) systems by classifying application domains and ML tasks, and identifying fairness-critical sensitive features early in the development lifecycle. In this study, we replicate and critically evaluate ReFAIR's methodology to assess its reproducibility and reliability. While ReFAIR achieved high reported F1-scores (97% for domain classification and 90% for ML task classification), our replication uncovered several inconsistencies in their methodology and limitations.**

Key issues included variability in results across different runs, lack of robust handling of noisy or mislabeled data, and incomplete documentation of sensitive feature recommendations. Our replication experiments confirmed strong performance for domain classification using BERT embeddings and XGBClassifier, with an F1-score of 98%. However, for multi-label ML task classification, while the Label Powerset method with GloVe embeddings and LinearSVC achieved an F1-score of 88.9% and a Hamming Loss of 0.392, these results showed sensitivity to preprocessing choices and random seeds. These inconsistencies suggest the need for more detailed guidelines and validation steps in the ReFAIR framework. Our findings emphasize the importance of transparency and reproducibility in fairness-aware frameworks and highlight areas where ReFAIR can be improved to better support practitioners.

***Index Terms*—Requirements Engineering, Fairness in Machine Learning, Reproducibility, Word Embeddings (TF-IDF, Word2Vec, GloVe, FastText, BERT), Domain Classification, Multi-Label Classification, Binary Relevance, Label Powerset, Classifier Chains, F1-Score, Hamming Loss, Sensitive Feature Identification**

I. Introduction

Fairness in Artificial Intelligence (AI) is a critical concern due to the increasing reliance on data-driven Machine Learning (ML) and Deep Learning (DL) systems. These systems often inherit biases present in datasets, training processes, and other components, resulting in fairness and reliability challenges. For example, researchers from KAUST identified a gender bias in the well-known generative model MidJourney [5], attributed to its reliance on outdated data and a narrow representation of specific nationalities [4]. This highlighted the importance of ethical AI, which lies in its ability to promote trust, accountability, and inclusivity in automated decision-making systems that impact various domains such as healthcare, finance, and social services. Ethical AI ensures that technology serves all individuals equitably, avoiding the amplification of societal biases and preventing harm to marginalized groups. Addressing these fairness challenges has led to the development of bias mitigation strategies, broadly categorized into two stages. The first, late-stage bias mitigation, includes techniques such as dataset resampling [3], ensemble learning [1], [3], and knowledge distillation [3]. These methods focus on modifying models or datasets during or after development to reduce bias. The second, early-stage bias mitigation, aims to identify and address biases before the development phase. This proactive approach can save significant time and effort by mitigating fairness issues early in the software life cycle.

ReFAIR [2] represents a pioneering framework in early-stage bias mitigation. It focuses on requirements engineering, specifically User stories (USs), to identify potential biases and recommend sensitive features that require attention. By integrating this analysis into the early stages of development, ReFAIR reduces pre-processing efforts and enhances software fairness from inception.

This replication study seeks to examine the robustness and reproducibility of ReFAIR's outcomes. Our goals are to identify potential gaps in the framework, ensure the correctness of its results, and provide recommendations for future improvements. This work contributes to validating ReFAIR's impact and exploring avenues for advancing fairness in AI-driven systems.

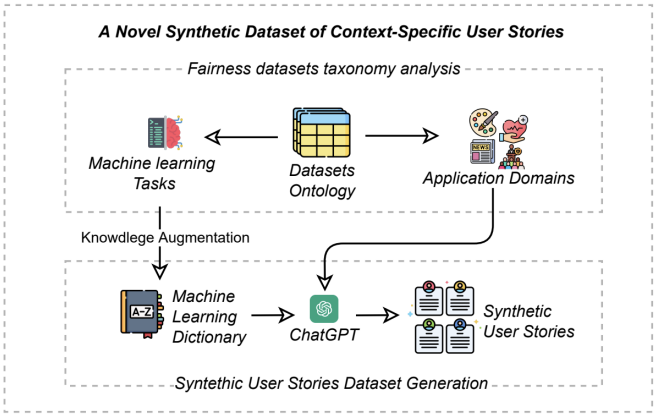

Fig. 1. Dataset generation process as described in ReFAIR [2].

II. Methodology

ReFAIR considered two aspects of their methodology: the generation of datasets, as shown in Fig. 1, and the treatment of bias in user stories. In our replication study, we will focus on the latter, assuming that their dataset is already fair.

A. Word Embedding

The first step in any Natural Language Processing (NLP) task is transforming text into numerical representations N size that ML models can process. ReFAIR employs several established embedding techniques: TF-IDF [6], Word2Vec [7],

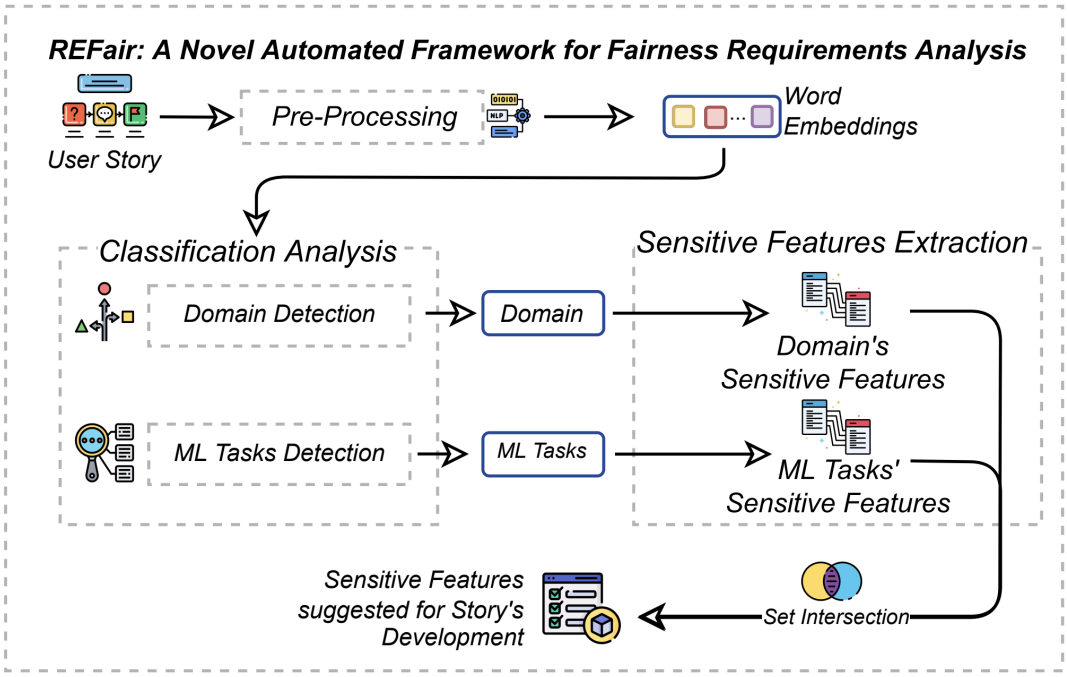

Fig. 2. Overview of the ReFAIR methodology, adapted from ReFAIR [2].

GloVe [8], FastText [9], and BERT [10]. These techniques vary in their approach to capturing word relationships and context. While traditional methods like TF-IDF and Word2Vec treat words more independently, contemporary approaches like BERT can generate context-aware representations by considering how words are used within sentences. A notable limitation in ReFAIR's implementation is its use of BERT's tokenizer without utilizing the model's pre-trained embedding layer. While tokenization converts text into numerical IDs, it doesn't capture the rich contextual information encoded in BERT's neural network. This design choice may impact the framework's ability to understand subtle semantic nuances in user stories. The choice of embedding technique significantly influences how well the system captures fairness-related concepts in different contexts. While simpler techniques might suffice for basic classification tasks, more sophisticated embeddings could better identify subtle fairness implications in user stories.

B. Domain Classification

The ReFAIR study approached domain classification as a multiclass classification problem, aiming to assign each user story to one of 34 predefined application domains. Under a supervised learning framework, every user story was mapped to a single domain from a fixed taxonomy. In their experiments, ReFAIR evaluated 25 different machine learning algorithms encompassing a range of methodologies. These included probabilistic classifiers (e.g., *Gaussian Naive Bayes*), methods grounded in information gain (e.g., *Random Forest*), and semi-supervised techniques (e.g., *Label Propagation*). Each model sought to determine decision boundaries that distinguish one domain class from another, using features extracted from the input data. During our analysis, we identified inconsistencies in the dataset used by ReFAIR. Specifically, certain domain labels contained typographical errors (e.g., *psychology* misspelled as *psycology* and *demography* as *demograpy*). These errors were not corrected in the original study, potentially affecting the quality of the classification results. Furthermore, there was no reported evidence of additional preprocessing efforts such as balancing class distributions or resolving ambiguous labels.

C. Classification of Machine Learning Tasks

ReFAIR employed a multi-label classification strategy for categorizing ML tasks. Unlike single-label classification, where each instance is assigned to only one class, multi-label classification allows user stories to be associated with multiple ML tasks simultaneously. For instance, a high-level classification task may involve multiple underlying models. The multi-label problem can be addressed using three main preprocessing strategies. The first is *Binary Relevance*, which transforms the problem into multiple binary classification tasks. The second is *Label Powerset*, which redefines the problem as a multiclass classification task over all combinations of labels, thereby capturing relationships between ML tasks. However, this approach can lead to a large number of possible classes, complicating model training. The third is *Classifier Chains*, which sequentially predicts labels by considering label dependencies. As in the domain classification setup, each user story was represented using vector embeddings before being mapped to the target ML tasks. ReFAIR evaluated

six machine learning models on these embeddings including Random Forest and Logistic Regression, to identify the best-performing approach.

In both the domain and ML task classification experiments, simple machine learning models were preferred over deep learning architectures. According to ReFAIR, these simpler models demonstrated strong performance, making more complex approaches unnecessary.

D. Sensitive Features Extraction

Sensitive features (SFs) extraction is a critical step in addressing fairness concerns within user stories. This process builds upon the prior classification of application domains and ML tasks, leveraging the Fabris dataset [11] as a foundational resource for identifying sensitive attributes. Each US is associated with a specific domain and potentially multiple ML tasks based on earlier classification stages. Using the base ontology, the framework systematically maps relevant sensitive features to each domain-task pair. These mappings identify attributes that may pose fairness risks, such as demographic characteristics, socioeconomic status, or other sensitive variables. To refine the extraction process, the framework computes the intersection of sensitive features independently identified for the domain and each associated ML task. This step ensures that only attributes relevant to both the domain and ML tasks are included in the final set of sensitive features. The resulting set represents actionable recommendations for practitioners, equipping them with a clear understanding of which attributes warrant attention to mitigate bias during subsequent development phases. To evaluate ReFAIR, the framework extracted a unique set of sensitive features by removing redundancies among sensitive features associated with different ML tasks. And then benchmarked using various techniques such as MoJo [12] distance which aims to measure the difference between the ideal set of sensitive features and the predictions made by ReFAIR, highlighting the framework's capability to recommend fairness-critical attributes effectively.

III. Experiments and Results

A. Experimental Setup

1) Embedding Models: Five-word embedding models are employed, using pre-trained models where applicable. Word2Vec utilizes a pre-trained model trained on Google News with 300-dimensional vectors. GloVe employs 100-dimensional vectors from the 6B dataset. FastText used a pre-trained model on English text with 300-dimensional vectors. TF-IDF is constructed from scratch, relying on corpus-specific term frequencies. BERT uses the base tokenizer from the Transformers library, excluding the first layer to align with ReFAIR. All embeddings are truncated to 100 features, with padding applied as necessary to ensure uniformity across datasets.

2) Experimental Procedure: For both the domain classification and ML task classification, the experiments were conducted using 10-fold cross-validation to ensure robust evaluation. All models were trained and tested on datasets derived from the five-word embeddings, and the average performance metrics across the folds were recorded. A fixed seed value of 42 was applied throughout to ensure reproducibility and eliminate randomness. Metrics used for evaluation included accuracy and F1-score for domain classification, and micro-averaged F1-score and Hamming Loss for ML task classification.

3) Domain Classification: The domain classification task involved 38 domains, with 4 erroneous classes (due to typos or redundancies) excluded during preprocessing. The task utilized the `LazyClassifier` module from the LazyPredict library to benchmark 25 machine learning models. The best-performing model and embedding combination was selected based on the highest average F1-score across the folds.

4) Machine Learning Task Classification: The ML task classification required handling multi-label data, making `LazyClassifier` unsuitable. Instead, six machine learning models—Random Forest, XGBoost, LinearSVC, Decision Tree, K-Nearest Neighbors (KNN), Logistic Regression, and Gaussian Naïve Bayes (GaussianNB)—were evaluated using `scikit-multilearn`. Label handling was performed using Binary Relevance, Label Powerset, and Classifier Chains methods. The best model was determined based on the highest micro-averaged F1-score across the folds.

5) Sensitive Features Recommendation: To evaluate sensitive feature recommendations, we used Oracle dataset as the ground truth and predictions generated by ReFAIR. The evaluation employed metrics such as **MoJo distance** [12] for structural similarity, **Levenshtein distance** [13] to account for typos, **Jaccard similarity** [14] for feature set overlap, and **Set overlap ratio** to quantify prediction accuracy.

The ReFAIR domain model was trained on 38 domain classes, addressing inconsistencies and typos by standardizing domain names, unlike the 34 classes mentioned in ReFAIR. The predictions were generated using the best performing models identified in RQ1 and RQ2. For each user story, the predicted domain was standardized, tasks were validated against domain-task mappings, and sensitive features were retrieved based on the validated tasks. The results included the predicted domain, tasks with their sensitive features, and unique sensitive features.

The analysis highlighted significant differences in results due to inconsistencies in the datasets and the expanded domain set. These findings underscore the importance of robust preprocessing and accurate domain-task mappings in improving sensitive feature recommendations.

B. Results

*1) **RQ1: The ReFAIR Application Domain Classification Performance**:* The domain classification task, being the simpler of the two, demonstrated strong results. As shown in Fig. 3, the best-performing model was XGBClassifier combined with BERT embeddings, achieving a 98.4% F1-score. Across all models, the results were highly consistent for each embedding, underscoring the critical role of word

TF-IDF			BERT			Word2Vec			FastText			GloVE		
Model	**F1-Score**	**Accuracy**	**Model**	**F1-Score**	**Accuracy**	**Model**	**F1-Score**	**Accuracy**	**Model**	**F1-Score**	**Accuracy**	**Model**	**F1-Score**	**Accuracy**
SVC	0.7999	0.802	XGBClassifier	0.984	0.984	LogisticRegression	0.970	0.971	LogisticRegression	0.951	0.951	CalibratedClassifierCV	0.961	0.962
CalibratedClassifierCV	0.7998	0.800	BaggingClassifier	0.981	0.981	CalibratedClassifierCV	0.970	0.971	CalibratedClassifierCV	0.950	0.951	LogisticRegression	0.960	0.961
ExtraTreesClassifier	0.7966	0.8	DecisionTreeClassifier	0.978	0.9777	LinearSVC	0.966	0.966	LinearSVC	0.949	0.948	LinearSVC	0.957	0.957

Fig. 3. Results for **Domain Detection Classifier**: F1-Score and Accuracy across different embeddings and models. The results represent the average of a 10-fold cross-validation.

embeddings and feature representation in domain classification. This consistency indicates that the task relies less on the specific machine learning model and more on the quality of the embedding.

Moreover, our results align closely with those reported by ReFAIR, showcasing the robustness of our approach despite variations in the number and types of domain classes, including erroneous ones. The minimal performance differences observed can primarily be attributed to variations in seed values, further confirming the stability of the results.

*2) **RQ2: The ReFAIR Machine Learning Tasks Classification Performance**:* The multi-label classification results reveal substantial variation in performance between the evaluated ML models, as illustrated in Fig. 4. The choice of multi-label handling method plays a pivotal role in achieving higher performance. The Label Powerset method consistently outperformed Binary Relevance across all word embeddings, as it captures label dependencies and relationships, unlike Binary Relevance, which treats labels independently.

The best F1-score was achieved by LinearSVC using the Label Powerset method combined with GloVe embeddings, reaching nearly 89%. In contrast, Hamming Loss, which measures prediction errors at the label level, showed complementary trends, with lower values indicating better model performance. Models that utilized Label Powerset generally achieved lower Hamming Loss, further highlighting its effectiveness in multi-label tasks.

Furthermore, our results align closely with those reported in the ReFAIR paper, demonstrating consistency and validating our methodology.

*3) **RQ3: Analysis of Sensitive Feature Prediction and Classification Metrics**:* From the Oracle dataset containing 12,401 USs we observed that more than half had empty sensitive features, with the Oracle ground truth indicating 60.13% of sensitive feature sets as empty. Similarly, our predictions identified 66% as empty sets, highlighting a significant issue in the original dataset. This discrepancy suggests that the high reported accuracy is not practically useful, as the majority of the classes do not contribute to identifying any sensitive features.

Despite these limitations, we achieved a perfect match rate of 77.18%, which includes both empty sets and exact matches for non-empty feature sets. This performance is lower compared to ReFAIR, which reported a 97% perfect match rate. For the remaining 22.82% of cases, our predictions deviated from the ground truth by one feature in 4.72% of cases, by two features in 4.25%, and by more than two features in 13.85%. These results indicate notable inconsistencies when compared to ReFAIR's reported outcomes.

In terms of distance metrics, our results yielded a Mean Jaccard Similarity of 0.8175, a Mean Set Overlap of 0.8180, a Mean MoJo Distance of 0.1822, and a Mean Levenshtein Distance of 0.8614. These metrics reflect moderate alignment with the ground truth but demonstrate room for improvement in feature recommendation accuracy.

Our domain classification task performed well, with only 58 out of 12,401 USs misclassified. The top five most common domain misclassifications included biology (16 times), literature (5 times), computer networks (4 times), education (4 times), and medicine (4 times).

However, the multi-label classification task proved more challenging, with an overall error rate of 34.21%. The top five most commonly predicted extra tasks were subset selection (386 times), resource allocation (376 times), pricing (313 times), representation learning (291 times), and spatio-temporal process learning (258 times). These misclassifications suggest the need for refinement in handling overlapping labels and improving the multi-label classification model's precision.

IV. Conclusion

ReFAIR represents an innovative approach to addressing fairness concerns during the early stages of ML development. By focusing on requirements engineering, it seeks to mitigate biases before they propagate into later stages of the ML lifecycle, thereby reducing preprocessing costs and enabling more efficient workflows. The framework's use of NLP and word embeddings to classify sensitive features from user stories highlights its potential to aid in bias-aware decision-making.

However, the findings from this replication study reveal several limitations that may affect ReFAIR's usability and replicability:

- **Unclear Methodology**: The framework lacks detailed documentation of intermediate processing steps, which hinders reproducibility.

TF-IDF			BERT			Word2Vec			FastText			GloVE		
Method+ML	F1-Score	Hamming Loss	Method+ML	F1-Score	Hamming Loss	Method+ML	F1-Score	Hamming Loss	Method+ML	F1-Score	Hamming Loss	Method+ML	F1-Score	Hamming Loss
LP_RandomForest Classifier	0.883	0.432	LP_DecisionTre eClassifier	0.856	0.576	LP_LinearSVC	0.825	0.392	LP_GaussianN B	0.750	0.384	LP_LinearSVC	0.889	0.391
LP_DecisionTreeCl assifier	0.821	0.363	LP_RandomFor estClassifier	0.834	0.573	LP_GaussianNB	0.698	0.285	LP_LinearSVC	0.718	0.287	LP_GaussianNB	0.679	0.281
CC_RandomForest Classifier	0.799	0.259	BR_DecisionTre eClassifier	0.772	0.356	BR_KNeighbors Classifier	0.596	0.279	LP_RandomFor estClassifier	0.701	0.273	BR_KNeighbors Classifier	0.624	0.275

Fig. 4. Results for **ML Task Detection Classifier**: F1-Score and Hamming Loss across different embeddings and models. The results represent the average of a 10-fold cross-validation.

- **Ambiguity in Pre-trained Models**: The selection and configuration of pre-trained models are not explicitly stated, leading to inconsistencies in implementation.
- **Insufficient Hyperparameter Tuning Details**: The absence of comprehensive descriptions of GridSearch parameters may result in suboptimal model performance.
- **Dataset Inconsistencies**: Variations in synthetic dataset naming conventions, particularly for domain names, reduce classification reliability.

Despite these challenges, ReFAIR demonstrates promising results in classifying machine learning tasks and identifying sensitive features, achieving strong performance when using embedding techniques such as BERT. This highlights the potential of early-stage bias mitigation in requirements engineering.

For future work, we recommend several enhancements to improve the robustness and applicability of the system. First, integrating large language models (LLMs), such as GPT-4 [15], can enhance the framework's ability to process complex user stories and capture domain-specific nuances. Additionally, incorporating Retrieval-Augmented Generation (RAG) techniques [16] will enable the dynamic retrieval of domain-specific knowledge, thereby improving the precision of sensitive feature recommendations. Standardized and publicly available benchmark datasets should also be developed and utilized to address inconsistencies and facilitate reproducibility. Finally, adding explainability features to the framework will provide insights into its recommendations, improving transparency and fostering user trust in the results.

In conclusion, ReFAIR offers a novel perspective on fairness requirements engineering and has the potential to address significant gaps in current bias mitigation practices. With further refinement and the adoption of modern AI methodologies, the framework can serve as a valuable tool for researchers and practitioners seeking to enhance fairness across the ML lifecycle.

REFERENCES

[1] Chen, Z., Zhang, J. M., Sarro, F., & Harman, M. (2022, November). MAAT: a novel ensemble approach to addressing fairness and performance bugs for machine learning software. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1122-1134).

[2] Ferrara, C., Casillo, F., Gravino, C., De Lucia, A., & Palomba, F. (2024, April). ReFAIR: Toward a Context-Aware Recommender for Fairness Requirements Engineering. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (pp. 1-12).

[3] Radwan, A., Zaafarani, L., Abudawood, J., AlZahrani, F., & Fourati, F. (2024). Addressing bias through ensemble learning and regularized fine-tuning. arXiv preprint arXiv:2402.00910.

[4] KAUST. (2024). KAUST Dear AI campaign targets gender bias in AI. Retrieved [Date Accessed], from https://www.kaust.edu.sa/en/news/kaust-dear-ai-campaign-targets-gender-bias-in-ai

[5] Midjourney. (2024). Midjourney (Version 6.1) [AI image generator]. Retrieved from https://www.midjourney.com/

[6] Karen Sparck Jones. "A Statistical Interpretation of Term Specificity and Its Application in Retrieval." Journal of Documentation*, vol. 28, no. 1, 1972, pp. 11–21.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space." Proceedings of the International Conference on Learning Representations (ICLR), 2013.

[8] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching Word Vectors with Subword Information." Transactions of the Association for Computational Linguistics, vol. 5, 2017, pp. 135–146.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.

[11] Alessandro Fabris, Stefano Messina, Gianmaria Silvello, and Gian Antonio Susto. 2022. Algorithmic fairness datasets: the story so far. *Data Mining and Knowledge Discovery*, 36(6), 2074–2152 (Sept. 2022). https://doi.org/10.1007/s10618-022-00854-z.

[12] Tzerpos, V., & Holt, R. C. (1999, October). MoJo: A distance metric for software clusterings. In *Sixth Working Conference on Reverse Engineering (Cat. No. PR00303)* (pp. 187–193). IEEE.

[13] Yujian, L., & Bo, L. (2007). A normalized Levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(6), 1091–1095.

[14] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles, 37*, 547–579.

[15] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

[16] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.