Three Ethical Dimensions of AI:
Fairness in Social Recommenders, Bias Detection in LLMs, and Privacy in NLP

by

Shera Potka
Bachelor (Digital Media and Computer Science), University of Cologne, 2022
Masters (Digital Media and Computer Science), University of Cologne, 2023

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

We acknowledge and respect the Lək̓ʷəŋən (Songhees and Xʷsepsəm/Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Three Ethical Dimensions of AI: Navigating Fairness in Social Network Recommender Systems, Bias in LLMs, and Data Privacy in NLP

by

Shera Potka

Supervisory Committee

Dr. Alex Thomo, Supervisor

(Department of Computer Science)

Dr. Venkatesh Srinivasan, Member

(Department of Computer Science)

Dr. Abdul Roudsari, Member

(School of Health Information Science)

ABSTRACT

This thesis investigates three foundational challenges in the development of responsible Artificial Intelligence (AI): fairness in social recommender systems, demographic bias in large language models (LLMs), and privacy-preserving techniques for Natural Language Processing (NLP). Though these problems differ in technical scope and application domain, they share a common thread: vector-based representations—embeddings of users, words, and tokens—fundamentally shape how AI systems behave, make decisions, and affect people. Across these three dimensions, this work introduces new methods for measuring, interpreting, and mitigating risk, offering solutions grounded in both empirical analysis and practical utility.

The first part of the thesis (Chapter 2) examines fairness in algorithmic link recommendation, with a focus on how structural minority communities—groups defined by network topology rather than identity—are represented in evolving social graphs. Standard recommenders tend to amplify popular users, reinforcing visibility gaps over time. We propose `MinWalk`, a fairness-aware algorithm that improves minority visibility while maintaining network stability. Simulations on real-world networks show that fairness- and diversity-aware algorithms vary widely in long-term impact, and that `MinWalk` offers a balanced, effective solution. This work underscores the importance of evaluating fairness dynamically and provides tools for designing more inclusive recommendation systems.

The second part (Chapters 3 and 4) turns to demographic bias in LLM behavior. We analyze gender and race associations in contextual embeddings from five leading models developed by OpenAI, Google, Microsoft, Cohere, and BGE. Using the SC-WEAT metric and clustering techniques, we show that stereotypical associations persist and are amplified in modern embeddings. We also examine how these biases appear in real-world applications, focusing on consumer product recommendations. Using prompt engineering and computational linguistics methods—including Marked Words, SVM classification, and dis-

tributional divergence—we find that LLMs generate demographically skewed suggestions that reinforce social stereotypes. These findings highlight the risks of bias in LLM outputs and offer concrete tools for auditing fairness in generative systems.

The final part (Chapter 5) addresses privacy in NLP, where the challenge lies in removing sensitive information from text without damaging meaning or fluency. Existing approaches either prioritize privacy but degrade text quality, or preserve fluency at the cost of weaker guarantees. To address this, we propose CluSanT, a flexible framework that uses token clustering and controlled replacement mechanisms to balance privacy and utility. Unlike prior methods, CluSanT retains strong privacy protection while producing more natural, semantically faithful text. We evaluate it using a range of metrics—including coherence, grammar, and semantic similarity—showing that it consistently improves over baselines on a legal benchmark dataset. Our results demonstrate that text sanitization can be both effective and intelligible to human readers.

Taken together, this thesis presents a unified perspective on ethical AI through the lens of embeddings. In social networks, language generation, and privacy-preserving NLP, vector representations are not neutral—they encode power dynamics, preferences, and access. By examining how these embeddings influence visibility, bias, and confidentiality, this work contributes both practical algorithms and conceptual frameworks for designing fair, inclusive, and trustworthy AI systems.

Keywords: Social Networks, Recommender Systems, Minorities, Bias, Large Language Models, Differential Privacy, Embeddings

Table of Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

It all began with one conversation. I still remember sitting across from Dr. Alex Thomo, unsure of where I stood or where I was going. I didn't have all the answers, I only had a deep curiosity, a stubborn work ethic, and a heart full of questions. I asked him if he thought I had what it took to pursue this path, to take on a PhD. He looked at me and said, simply and without hesitation, "Yes. I believe in you." That one moment changed the course of my life.

What followed was a journey I could have never imagined. I moved from Germany to Canada far from home, family, and everything familiar. I entered a new world filled with opportunity, but also steep hills to climb. There were days when I was balancing teaching, research, and instructing just to make ends meet.

To Dr. Thomo—thank you for seeing something in me that I didn't yet see in myself. Your mentorship has been a steady light through every dark tunnel, and your support gave me the courage to keep moving forward. You didn't just guide my research, you believed in the researcher I could become.

To my family, you have been my constant source of strength. Across oceans and time zones, your love never wavered. You believed in me when I had nothing but exhaustion to offer. You reminded me of who I am when I started to forget. Every word in this dissertation carries your faith and your sacrifices. I hope this achievement makes you proud, because it belongs to you too. This story is not just mine. It belongs to the people who stood quietly beside me, who lifted me up with kind words, shared meals, thoughtful messages, and silent prayers. I wish I could name every single person who played a role in this chapter of my life. Please know—if you were part of this journey in any way, I am deeply, deeply grateful. I started this path with a question. I end it with a story—a story of resilience, hope, and the power of people who choose to believe in each other.

DEDICATION

My loved ones.

Chapter 1

Introduction

Artificial Intelligence (AI) systems have become central to digital life, shaping communication, decision-making, and access to information. While the power and reach of AI continue to expand, so do concerns over fairness, bias, and privacy—issues that cut across the technical and social dimensions of these systems. This thesis addresses these concerns by examining how algorithmic systems behave in practice: how they treat different demographic groups, how they recommend connections and products, and how they handle private information. We focus on three key areas: fairness in social network recommender systems, demographic bias in large language models (LLMs), and privacy-preserving natural language processing (NLP). These areas are often treated separately, but we argue they are deeply connected. All three reflect how AI systems represent and manipulate human data and how those representations impact people's lives.

1.1 Fairness in Social Network Recommender Systems

Social networks profoundly influence how individuals form communities and receive information. While early network growth was driven by homophily and preferential attachment, today's social networks are increasingly shaped by link recommendation algo-

rithms. These systems, designed to optimize engagement, often prioritize popular nodes and reinforce echo chambers. This leads to filter bubbles, reduced exposure to diversity, and—critically—the marginalization of minority groups.

Our first line of research investigates the visibility of structural minorities in social network recommendation systems. Unlike prior work that used synthetic data or defined minorities based on static attributes (e.g., gender, race), we define minorities structurally, through dense subgraphs that emerge naturally in real-world networks. These structural communities can represent tightly connected professional groups, local interest clusters, or niche social circles that would otherwise be overlooked by standard algorithms.

We analyze how different link recommendation algorithms–especially those designed for fairness or diversity–impact the visibility of these structural minorities. Our findings show that while some algorithms improve visibility, others remain indifferent or amplify visibility to a degree that could alienate majority users. To strike a balance, we introduce MinWalk, a new algorithm designed to enhance minority visibility over time without destabilizing user engagement. Through extensive simulations, we also examine popularity bias and temporal feedback effects to evaluate long-term impacts. This work contributes to a more varied understanding of fairness in algorithmic design, showing that well-intentioned fairness algorithms may still fall short if they ignore structural group dynamics.

1.2 Demographic Bias in Large Language Models

Large Language Models (LLMs) are rapidly becoming foundational technologies in search, communication, and decision-making systems. However, as their influence grows, so does concern over the demographic biases they encode and propagate. In this line of research, we explore two facets of this problem: how biased associations are embedded at the word level within LLMs, and how these biases surface in applied tasks such as consumer product

recommendation. Together, these investigations reveal the depth and reach of representational bias in modern language systems.

1.2.1 Word Embedding Bias in LLMs

Modern LLMs from OpenAI, Google, Microsoft, and others rely on word embeddings to encode semantic meaning. These embeddings, learned from massive text corpora, capture not just linguistic patterns but also social context—including biases tied to gender, race, and identity. This part of our research examines how these embedded associations manifest in state-of-the-art LLMs and what they reveal about the models' internal representations.

We focus on identifying and quantifying demographic biases in LLMs, particularly those related to gender and race. Building on earlier studies like Caliskan et al.'s Word Embedding Association Test (WEAT), we extend the scope by analyzing embeddings from modern contextual models, including OpenAI, Cohere, Google, Microsoft E5, and BGE. We study how word frequency, thematic clustering, and semantic associations contribute to bias across 100,000 of the most frequent tokens.

We use methods such as k-means clustering, t-SNE visualization, and cosine similarity to identify and group biased concepts. Our work also surfaces how biased embeddings manifest in real-world domains, especially in Big Tech and higher education. This deep, quantitative analysis offers new tools for detecting bias in LLM embeddings at scale, and new insights into how model architecture and training data shape social meaning.

1.2.2 LLM-Driven Product Recommendation Bias

Beyond the embeddings themselves, LLMs are now used in more complex, task-oriented systems, including product recommendation engines. These systems make seemingly personalized suggestions, but under the hood, they can encode and propagate subtle forms of bias. While past research has emphasized overt stereotypes, we focus on implicit bias—the

kind that appears in small, linguistic patterns or asymmetric associations between demographic groups and product categories.

To analyze this, we use prompt engineering to elicit demographic-specific product recommendations from LLMs. We apply three computational techniques: marked word analysis, support vector machines (SVMs), and Jensen-Shannon divergence, to identify systematic differences in language and product placement across gender and race groups. Our findings reveal that LLM-generated recommendations frequently diverge in ways that reinforce societal hierarchies—suggesting, for instance, less prestigious or lower-value items for certain groups.

This work bridges a key gap in the literature: applying fine-grained linguistic analysis to practical, high-impact use cases. It also provides tools for diagnosing and mitigating implicit bias in black-box LLM systems—a step toward building inclusive AI systems that align with ethical goals.

1.3 Privacy-Preserving Text Sanitization in NLP

As NLP systems are deployed in sensitive domains–healthcare, law, education–the need for strong privacy protections grows. Differential Privacy (DP) offers a principled approach, but applying it to natural language is not straightforward. Existing methods like SanText and CusText face a critical tradeoff: stronger privacy often comes at the cost of garbled, low-utility output; more fluent output may violate privacy guarantees.

Our third research thread tackles this challenge with CluSanT, a framework for differentially private and semantically coherent text sanitization. CluSanT introduces three components–token clustering, cluster embeddings, and token sanitization–to offer a tunable balance between utility and privacy. We use LLM-derived embeddings to drive the clustering and selection processes, ensuring replacements are both privacy-preserving and

semantically appropriate.

Unlike prior methods that randomly sample replacements based on distance metrics, CluSanT operates over semantically meaningful clusters of tokens, increasing the likelihood of coherent output. We show that CluSanT not only improves utility metrics and text coherence but also satisfies metric local differential privacy guarantees across a range of parameter settings. Our framework generalizes existing approaches and allows users to fine-tune the privacy/utility tradeoff to meet different application needs.

1.4 A Unified Perspective

While each line of research addresses a distinct problem, they are connected by a shared emphasis on representation—who is visible in the system, how they are portrayed, and whether their data is treated responsibly. All three projects rely on vector-based embeddings as the common language of AI systems. In our recommender system work, users are embedded based on random walks, effectively treating user paths as words in a social vocabulary. In our bias analysis, we use word and phrase embeddings to measure semantic proximity and detect patterns. In our privacy work, we sanitize sensitive text using embedding-based clustering to preserve coherence.

Together, these threads offer a cohesive view of ethical AI design. Fairness in social recommendations, demographic bias in LLM behavior, and privacy in NLP may target different applications, but they all reveal how algorithmic decisions can reflect and reinforce social inequalities. Embeddings—used to rank users, associate concepts, and generate language—form the underlying mechanism where these dynamics emerge. By developing methods to audit and evaluate how embeddings encode bias, amplify disparity, or compromise privacy and utility, this work contributes to building more accountable, inclusive, and trustworthy AI systems.

1.5 Contributions

To summarize, this thesis explores three major facets of ethical AI: fairness in social network recommender systems, demographic bias in large language models, and privacy in natural language processing. Each of these areas is developed in depth across multiple chapters and is supported by peer-reviewed publications corresponding to the core contributions.

Fairness in Social Network Recommendations: Chapter 2 studies the long-term visibility of structural minority communities in link recommendation systems. Unlike prior work that focuses on attribute-based or synthetic groups, our work emphasizes real-world structural communities formed through connection density. We propose *MinWalk*, a fairness-aware algorithm that enhances minority visibility while mitigating backlash effects and reducing popularity bias. Our research is published in the following article:

- Shera Potka, Isla Li, Jason Kepler, and Alex Thomo. *Enhancing Structural Minority Visibility in Link Recommendations*. MEDES 2024 (16th International Conference on Management of Digital EcoSystems). **Best Paper Award**.

Bias in LLM Word Embeddings: Chapter 3 investigates how modern LLMs embed gender and race biases into their word representations. Extending the SC-WEAT framework, we evaluate 100,000 words across several LLMs and identify thematic clusters of biased associations. This chapter offers the first large-scale analysis of both gender and race bias in contextual LLM embeddings. Our research is published in the following article:

- Poomrapee Chuthamsatid, Shera Potka, and Alex Thomo. *Word Embedding Bias in Large Language Models*. I-SPAN 2025 (17th International Symposium on Pervasive Systems, Algorithms, and Networks).

Implicit Bias in LLM Product Recommendations: Chapter 4 extends the bias analysis to applied LLM tasks, focusing on consumer product recommendations. Using prompt

engineering and linguistic analysis (e.g., marked words, SVMs, JSD), we uncover demographic disparities in model outputs, even in ostensibly neutral recommendation tasks. Our research is published in the following article:

- Ke Xu, Shera Potka, and Alex Thomo. *Gender and Race Bias in Consumer Product Recommendations by Large Language Models*. AINA-2025 (39th International Conference on Advanced Information Networking and Applications).

Privacy-Preserving Text Sanitization: Chapter 5 focuses on privacy in NLP, addressing the limitations of current differentially private text sanitization methods. We propose *CluSanT*, a novel MLDP-compliant framework that improves semantic coherence while maintaining strong privacy guarantees. The framework generalizes prior methods and allows for tunable trade-offs. Our research is published in the following article:

- Ahmed Musa Awon, Yun Lu, Shera Potka, and Alex Thomo. *CluSanT: Differentially Private and Semantically Coherent Text Sanitization*. NAACL 2025 (Annual Conference of the North American Chapter of the Association for Computational Linguistics).

Chapter 2

Fair and Diverse Link Recommendations: Minority Visibility Impact

2.1 Introduction

Social networks, impacting everything from collaboration to well-being and societal outlooks [75], have evolved from being shaped by homophily and preferential attachment [61] to being influenced by link recommendation algorithms [6]. These algorithms tend to favor popular choices [17], leading to filter bubbles and echo chambers [47] and heightening societal polarization. Additionally, they risk marginalizing minority voices [41, 86, 40], thus perpetuating stereotypes and limiting diversity [32].

A set of recent works, [80, 81, 42, 26, 43], have concentrated their focus towards understanding the temporal effects of link recommendation algorithms in social networks. This temporal perspective is important as the dynamics of social networks evolve over time, and the long-term consequences of algorithmic interventions can be profound. In particular, the

studies by Fabbri et al. and Ferrara et al. [42, 43] model the feedback loop of user interactions with link recommenders, affecting network structure and *minorities*, with Ferrara et al. offering a novel metric for minority visibility based on PageRank and tools to measure popularity bias and cohesiveness in social networks. However, these works come with limitations which we outline as follows: (1) they rely on synthetic data for their studies, (2) they do not consider natural minorities that emerge organically based on their connection density, and (3) they only examine traditional link recommender algorithms ignoring newer, fairness- and diversity-aware algorithms.

In our study, we examine the visibility of minority groups in *real-world* social networks– a shift from the *synthetic data* reliance seen in [42, 41, 26, 43]. Contrary to the attribute-based community definitions in works like [80, 41] (e.g. men vs. women), or homophilic community definitions in [42, 26, 43], our research is grounded in *structural communities* (based on link density) within networks, aiming for a more accurate reflection of network clusters and minority communities. Structural communities can reveal hidden groups that we might miss with traditional methods based on attributes like gender or age. For example, in a social network, users who interact frequently might form a structural community even if they do not share obvious attributes. These groups could be tightly-knit professional circles, niche hobby clubs, or local interest groups. By focusing on these structural communities, we make sure fairness extends to all kinds of minority groups, not just the ones that fit into conventional categories.

A key feature of our work is the use of *fairness and diversity-aware recommendation algorithms*, which is in contrast to works such as [42, 26, 43] that only consider classical recommendation algorithms. In light of recent developments in fairness and diversity-aware recommendation algorithms (cf. [74, 86, 76, 54]), our study is the first to analyse these algorithms with respect to minority visibility and popularity bias. This is important because, although these algorithms have been designed with fairness and diversity in mind,

they in fact pursue different objectives. These objectives, however, do not necessarily prioritize the visibility of minorities over time as users engage with the algorithm. Therefore, in this work, we aim to examine how the various fairness and diversity objectives of previous works align with the goal of increasing minority visibility over time.

Contributions: We make the following contributions in this chapter. (1) We demonstrate that algorithms designed with fairness or diversity considerations show varied impacts on the visibility of structural minorities according to algorithmic ranking. While some of these algorithms appear indifferent to structural minorities, others increase their visibility significantly, potentially to a level that could risk backlash from the majority. This extreme visibility enhancement might lead to majority users abandoning the recommendation algorithm altogether. (2) We introduce a new algorithm, MinWalk, designed to effectively enhance the visibility of minorities in a balanced manner, aiming to minimize network disruptions and reduce the potential for significant dissatisfaction among the majority user base. (3) We also analyze the popularity bias of the on-focus algorithms, examining their temporal effects over multiple iterations. This involves assessing whether these algorithms inadvertently favour more popular nodes at the expense of lesser-known ones.

2.2 Related Works

Impact of Recommendation Algorithms on Communities. Several studies have explored the impact of recommender systems on network communities, offering valuable insights but also presenting certain limitations.

Cinus et al. [26] investigate the effect of recommender systems on echo chambers and polarization using Monte Carlo simulations. While innovative, their study is limited to synthetic datasets, homophilic communities, and use of classical recommender algorithms. We note that homophilic communities are defined by shared attributes, whereas structural com-

munities are characterized by dense connections among members. Espín-Noboa et al. [40] explore how preferential attachment and homophily contribute to inequality in network-based ranking algorithms, but their study is limited to the effects of a single attribute and a specific network model.

Fabbri et al. [41] look at how similarity in choices or traits (homophily) can make minority groups more visible in recommendation systems, more so than the group's size. They provide a static, single-round analysis, which does not consider the changes over time or structural communities. Their subsequent study [42] suggests that these algorithms could help small homophilic groups become more noticeable in the long run. However, this work also does not address how structural communities evolve, and it is limited to a one-iteration analysis.

Of particular relevance to our discussion is the work by Ferrara et al. [43]. They define minority visibility in networks and introduce a novel metric based on the PageRank algorithm to quantifiably measure it. Furthermore, they propose methods to capture both popularity bias and network cohesiveness using established network metrics.

Stoica et al. [80] and Su et al. [81] study biases in social media link recommendation algorithms. Stoica et al. [80] examine Instagram's "glass-ceiling" effect due to link recommendation algorithms, which may restrict the network growth of groups such as women or men. Su et al. reveal a "rich get richer" trend on Twitter, where popular users gain more from these algorithms, leading to amplified visibility for already high-profile users.

Fainess-aware link recommendation algorithms. Several studies have notably emphasized fairness and diversity in recommendation algorithms, primarily focusing on random walk-based methods [74, 76, 54, 86]. Rahman et al. [74] developed Fairwalk, enhancing the node2vec embedding approach [48] by integrating fairness metrics like statistical parity to reduce bias in friendship recommendations while maintaining utility. Saxena et al. [76] introduced NodeSim, a network embedding method that accounts for community

structures to enhance both intra and inter-community link prediction in social networks, and promoting diverse link predictions in the process. In a similar vein, Khajehnejad et al. [54] proposed CrossWalk, a method that biases random walks to cross group boundaries, thus promoting fairness and diversity in these walks. Further, Tsioutsiouliklis et al. [86] modified the Pagerank algorithm to create fairness-sensitive personalized Pagerank to meet a 'universal personalized fairness' criterion.

2.3 Networks, Communities, and Link Recommendation

In this work, we consider both directed and undirected network graphs. A graph G is defined as a pair (V, E), where V represents the set of vertices (or nodes) and E represents the set of edges (or links). Vertices are denoted by letters such as u, v, and edges are represented as pairs (u, v). For undirected graphs, each edge is treated as two directed edges in opposite directions. Consequently, we do not make a distinction between directed and undirected graphs in our analysis.

Structural Communities. Social networks are typically sparse graphs, where the number of edges is within a constant factor of the number of vertices. However, the nodes in these graphs often cluster in regions of higher density, which are referred to as structural communities in network analysis literature (cf. [46, 52, 73]). The main characteristic of these communities is that they have more connections among their members than with nodes outside the community. Structural communities play a crucial role in understanding and optimizing social dynamics [68]. They optimize network connectivity and information flow [7, 39], aid in disease modeling [24, 50], enhance recommendation systems [2, 55], foster professional collaboration [12], and facilitate effective marketing strategies [75], while providing valuable data for sociocultural research [46].

In this work, we utilize two well-known algorithms to discover structural communities:

the Louvain algorithm [9] for undirected graphs, and the Leiden algorithm [85] for directed graphs. The Louvain algorithm detects high-quality community structures by optimizing modularity in a hierarchical manner [9]. On the other hand, the Leiden algorithm, an improvement over the Louvain method, focuses on refining community detection for directed networks [85]. The output of these algorithms is a partitioning of the nodes of the graph into structural communities, i.e. each node is assigned to a community. In our approach, the minority group within each network is identified by sorting the nodes by their community size and selecting the smallest $r\%$ of communities (e.g., 15%). These are then combined to form the minority group, denoted by M. The remaining nodes are considered to be the majority group, denoted by J. We have $M \cup J = V$.

Link recommendation Link recommendation algorithms utilize the network structure to suggest top-k matches for each node u. When node u accepts these recommendations, it changes the network structure, creating a feedback loop. This altered structure is then analyzed by the algorithm for new recommendations, thus continuing the cycle.

Central to most recommendation algorithms is the concept of node similarity. Nodes are typically recommended to connect with others that are most similar to them with respect to their one- and multi-hop friends. However, the definition and computation of similarity can vary. A prominent approach, Node2Vec [48], involves performing random walks starting from each node. These walks generate "sentences," with "words" representing node IDs encountered during the walk. The sentences are then embedded into a multidimensional space, assigning a vector to each node. The similarity between nodes is inferred from the similarity of their corresponding vectors–the closer two vectors are in this space, the more similar the nodes they represent are considered.

Fairness and diversity-aware algorithms bias random walks for more equitable recommendations, evolving networks towards greater diversity and/or fairness.

2.4 Algorithms

In this section, we provide a more detailed description of the fairness and diversity-aware algorithms for link recommendation [74, 76, 54]. In our work, we restrict the scope of our analysis to random-walk-based algorithms due to their popularity and effectiveness in capturing the structural properties of graphs. Our aim in this chapter is twofold: to present a comparative study of state-of-the-art fairness and diversity-aware algorithms for link recommendation using random walks, focused on the visibility of minorities, and to introduce our new algorithm, Min-Walk, designed to enhance minority visibility in a balanced way.

2.4.1 FairWalk

FairWalk, introduced by Rahman et al. in [74], offers a fairness-oriented approach to generating random walks, addressing the representation bias often seen in standard methods like node2vec. The proposed technique modifies the traditional random walk procedure by first categorizing neighboring nodes into groups according to sensitive attributes. Unlike the conventional method where any neighbor might be chosen for the next step in the walk, FairWalk guarantees that each group–regardless of its size–has an equal probability of being selected. From the chosen group, a node is then randomly picked to proceed with the walk. In our evaluation, we use two groups within our setting: the minority and the majority, as the basis for applying this method.

2.4.2 NodeSim

NodeSim, introduced by Saxena et al. in [76], enhances random walks by promoting diversity through steps across communities. In contrast to conventional methods like node2vec, where the transition probability from node u to node v is $\frac{1}{\deg(u)}$, NodeSim incorporates node similarity and community structure into its random walk process. The unnormalized

probability p_{uv} for moving from node u to node v is defined as $\alpha \cdot (\text{Sim}(u,v) + \frac{1}{\deg(u)})$ if u and v are in the same community and $\beta \cdot (\text{Sim}(u,v) + \frac{1}{\deg(u)})$ if they are in different communities, where $\text{Sim}(u,v)$ represents the Jaccard similarity of their neighbor sets. The parameters α and β play the role of guiding the random walker: a higher value of α encourages the walker to sample similar nodes within the same community, while a higher value of β motivates the walker to explore nodes outside the community of a node.

2.4.3 CrossWalk

CrossWalk, introduced by Khajehnejad et. al. in [54], aims to strengthen network connectivity across diverse groups by weighting edges more heavily near or across group boundaries.

For each node v within a graph, a quantifiable metric, denoted as $m(v)$, is established to assess its proximity to nodes of different groups. $m(v)$ is defined as the expected count of encounters with nodes from other groups during r random walks of length d that start from v: $m(v) = \frac{1}{r \times d} \sum_{j \in [r]} \sum_{u \in W_j^v} I[l_v \neq l_u]$ where, W_j^v represents the set of nodes visited during the j^{th} random walk from node v, and $I[l_v \neq l_u]$ is an indicator function that assumes a value of 1 when the group label l_v of the starting node v is not equal to the group label l_u of the node u encountered during the walk, and 0 when they are the same. Consequently, this measure cumulatively evaluates the interaction of node v with nodes from other groups across all walks, providing an aggregate indicator of its connectivity to diverse groups within the graph. Nodes adjacent to group boundaries with a diverse label mix in their proximity gain a higher m value, leading to an inclination of reweighted random walks toward these boundary nodes.

Edge weights are then introduced, which are biased towards promoting diversity with different weights for same-group $(1-\alpha) \times m(u)^p$ and different-group $\alpha \times m(u)^p$ connections, controlled by α and p. While CrossWalk enhances diversity, empirical results show it

can sometimes overemphasize minority communities in link recommender systems. To address this, we introduce a new algorithm, MinWalk, which seeks a more balanced approach to promoting cross-boundary walks.

2.4.4 MinWalk

We now introduce our algorithm, MinWalk (Minority Walk), specifically designed to weigh edges in a manner that enables random-walk-based link recommender algorithms to generate walks with an emphasis on minority communities.

MinWalk operates by adjusting the edge weights of a specific small set of majority nodes, thereby only moderately interfering with the graph's original structure. Unlike CrossWalk, which generally improves minority visibility but often goes overboard in this goal (as our experiments show), MinWalk is more targeted and less disruptive. The primary goal of MinWalk is to better align the proportion of minority nodes within the top $t\%$ of PageRank (PR) values to the graph's initial minority ratio (denoted as μ). This is achieved by selectively adjusting the weights of majority nodes that are only marginally within the top $t\%$ bracket of PR values.

Adjustment Strategy. Consider a hypothetical graph with 1000 nodes, $t = 10\%$, and where the minority ratio $\mu = 15\%$. Ideally, this ratio should be mirrored in the top 10% of PR scores. If the existing minority ratio in this bracket is only 5%, MinWalk steps in to make adjustments. It targets the lower-ranked majority nodes in this top bracket, modifying the weights of the bottom 10 (from 95 to 85) of these majority nodes, to achieve the desired 15% minority representation.

In MinWalk, edge weight adjustments are made exclusively to majority nodes, denoted as J, while minority nodes, denoted as M, remain unchanged. The algorithm retains the closeness calculation from CrossWalk, emphasizing nodes at the majority periphery. The closeness value $m(v)$ for each node v is now specifically determined by the proportion of

minority nodes encountered during random walks.

Algorithm 1 MinWalk: Minority-enhanced edge weighting

Require: Graph $G = (V, E)$, minority set $M \subset V$, majority set $J \subset V$ ($M \cup J = V$), top ratio t of interest in PR ranking.

Ensure: Weights $w_{vu}, \forall (v, u) \in E$.

$n = |V|$, $\mu = |M|/|V|$ ▷ n is the number of nodes, μ is the minority ratio.

$H = \{u \in J \mid rank(u, V) \leq t \cdot n\}$ ▷ H is the set of J nodes in the top t% of PR ranked nodes.

$K = \{u \in J \mid rank(u, J) \leq t \cdot n \cdot (1 - \mu)\}$ ▷ K is the set of H nodes to stay in the top t% of rankings.

$L = H \setminus K$ ▷ L is the set of J nodes that will have their edges reweighted.

for $v \in V$ **do**

    Run r random walks $W_j^v, j \in [r]$ rooted at v.

    $m(v) = \frac{1}{r \cdot d} \sum_{j \in [r]} \sum_{u \in W_j^v} I[u \in M]$ ▷ Closeness of J node v to minority M.

    $IS(v) \leftarrow \frac{|N_v \cap M|}{|N_v|}$ ▷ IS (influence score) is the ratio of the M nodes in N_v (neighborhood of v).

for $v, u \in V$ **do** ▷ Initial weights: higher weight for nodes with more M neighbors.

    $w_{vu} = \frac{IS(v) + IS(u)}{2}$

for $v \in L$ **do** ▷ Reweighing edges.

    $Y_v = \sum_{u \in N_v \cap J} w_{vu} \cdot m(u)$ ▷ Normalization quantity for reweights of edges incoming to a J node.

    $Z_v = \sum_{u \in N_v \cap M} w_{vu} \cdot m(u)$ ▷ Normalization quantity for reweights of edges incoming to a M node.

    if $v \in L$ **then** ▷ Valid Majority nodes

        for $u \in N_v \cap J$ **do** ▷ Edges within J.

            $w_{vu} = w_{vu} \cdot \frac{m(u)}{Y_v}$ ▷ Edges going to nodes closer to M get higher weight.

        for $u \in N_v \cap M$ **do** ▷ Edges from J to M.

            $w_{vu} = w_{vu} \cdot \frac{m(u)}{Z_v}$ ▷ Edges going to M nodes get higher weight.

Details. To aid in understanding MinWalk, we present annotated pseudocode in Algorithm 1. We set $n = |V|$ and the minority ratio $\mu = |M|/|V|$. The algorithm then focuses on identifying key subsets within the majority set J. Firstly, it defines H as the set of nodes in J that rank in the top $t\%$ of PageRank (PR) ranked nodes within the entire set V. Subsequently, it refines this to a subset K, comprising nodes in H that remain in the top $t\%$ after adjusting for the minority ratio. This leaves us with the set $L = H \setminus K$, which consists of majority nodes targeted for edge reweighting.

For each node $v \in V$, the algorithm conducts r random walks to calculate $m(v)$, a measure of the closeness of a majority node v to the minority set M. It also computes the Influence Score $IS(v)$, defined as the ratio of minority neighbors to the total number of neighbors of v. The initial weights for the edges (v, u) are set as $w_{vu} = \frac{IS(v)+IS(u)}{2}$, emphasizing connections to nodes with a higher proportion of minority neighbors.

The reweighting process is applied to nodes in L. For each $v \in L$, normalization quantities Y_v and Z_v are calculated, summing the weighted closeness measures $m(u)$ for majority and minority nodes, respectively. The algorithm then adjusts the weights of the edges. For edges within the majority set J, the weights are modified as $w_{vu} = w_{vu} \cdot \frac{m(u)}{Y_v}$, and for edges from the majority to the minority set M, as $w_{vu} = w_{vu} \cdot \frac{m(u)}{Z_v}$. This reweighting process selectively enhances the influence of edges that connect to the minority set, or are within the majority set. This, in turn, amplifies the representation of minority nodes in the random walks performed by a random-walk-based link recommender system.

2.4.5 Universally-Fair Personalized PageRank

Finally, an algorithm that does not explicitly generate traces from random walks (but still based on them) is Universal Personalized Fairness in PageRank (u-ppr), introduced by Tsioutsiouliklis et al. in [86]. It modifies the standard personalized PageRank algorithm to prioritize equitable treatment across nodes. Their adjustment results in a personalized probability vector $\text{PR}_v(u)$ that emphasizes fairness by distributing probabilities evenly among groups based on their proportions.

2.5 Experiments

Network Evolution Methodology. In our study, we begin with a network graph G and a given recommendation algorithm A. For each node u in G, algorithm A provides a top-1

link recommendation, identifying the most suitable node v for u to connect to. Following the approach of previous studies [43, 26], we then establish an edge between u and v, and additionally, we remove a random edge incident to u. In the case of a directed graph, this involves adding a directed edge from u to v and then removing a random outgoing edge from u. This strategy of adding and removing edges is designed to prevent an excessive increase in the network's edge density. We focus on maintaining constant edge density, as our evaluation metrics are sensitive to it.

By removing one connection for each new one formed, we make sure the network's density stays stable, allowing us to attribute any observed changes solely to the recommendation algorithm A, without the influence of increased total connections. This methodology aligns with social theories that suggest individuals have a finite capacity for communication, limiting the number of active ties they can maintain [66]. The issue of whether network users will accept the top recommendations for edge additions relates to the "acceptance policy". However, studies [26, 41] suggest that this policy has minimal impact on the evolution of the network. The above process is repeated for 30 iterations to assess the recommendation algorithm's effects over time.

Structural Minorities. In order to determine structural minorities in undirected and directed graphs, we utilize the Louvain [9] and Leiden [85] algorithms, respectively, for community detection. The minority group within each network is identified by sorting nodes by community size and selecting the smallest $r\%$ of communities. In the results we show here, we used an $r\%$ value of 15%, although we also tested values of 5%, 10%, 20%, and 25%, all of which exhibited similar behavior.

Metrics. The first metric we evaluate is 'minority visibility,' as introduced by [43], which assesses the prominence and network position of minority nodes following the implementation of recommendation algorithms. This metric specifically measures the proportion of minorities within the top 10% of nodes ranked by PageRank. It is essential to determine

whether the algorithms enhance or diminish the representation and influence of minority nodes.

The second metric we evaluate is the Gini Coefficient, a measure of inequality in the distribution of connections (degrees) among nodes. A higher value indicates a network where connections are concentrated among a few nodes, which could lead to the marginalization of minority nodes. This metric is useful for assessing how much recommendation algorithms either contribute to or alleviate inequalities in network connections.

Name	**Nodes**	**Edges**	**Clust. Coeff.**	**Type**
congress	475	13,289	0.2242	Directed
email-eu	1,005	25,571	0.3657	Directed
wiki-vote	7,115	103,689	0.0816	Directed
facebook	4,039	88,234	0.6055	Undirected

Table 2.1: Datasets used in our experiments

Datasets. We utilize datasets from the Stanford Large Network Dataset Collection (`https://snap.stanford.edu`), including: **Congress** (Twitter interactions of the 117th U.S. Congress), **EU-Email** (email interactions within a European research institution), **Wiki-Vote** (Wikipedia voting data until January 2008), and **Ego-Facebook** (Facebook 'circles' or 'friends lists'). Details on nodes, edges, and clustering coefficients are provided in Table 2.1.

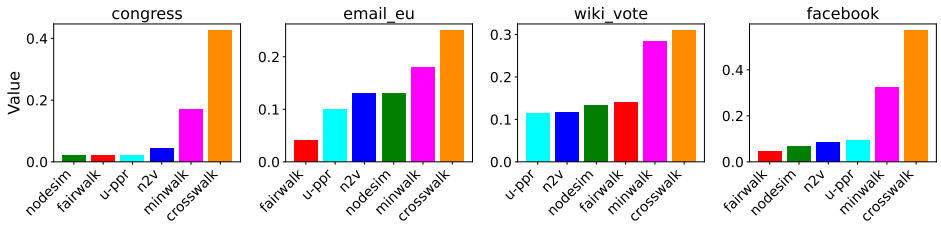


Figure 2.1: Visibility after 30 iterations for minority size of 15%. In all the charts we observe that crosswalk increases the visibility well above 15%, sometimes to above 50% (see 'facebook'). A more balanced approach is our 'minwalk' algorithm which increases the visibility of minorities more moderately or keeps it close to 15%.

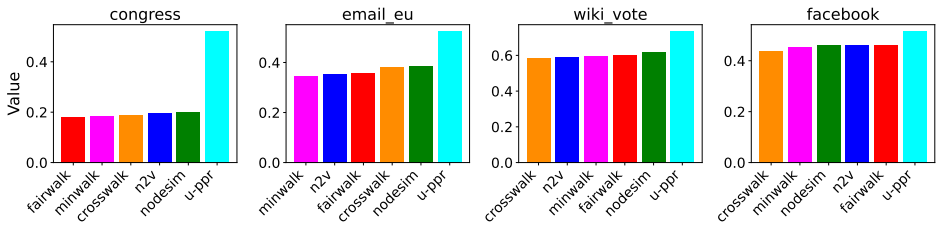


Figure 2.2: Gini after 30 iterations for minority size of 15%. The greater the Gini coefficient, the more imbalanced or unequal the distribution of degrees. 'u-ppr' leads to most imbalance. 'minwalk' tends to result in Gini coefficients at the lower end, indicating its propensity to evolve the network towards a more balanced state.

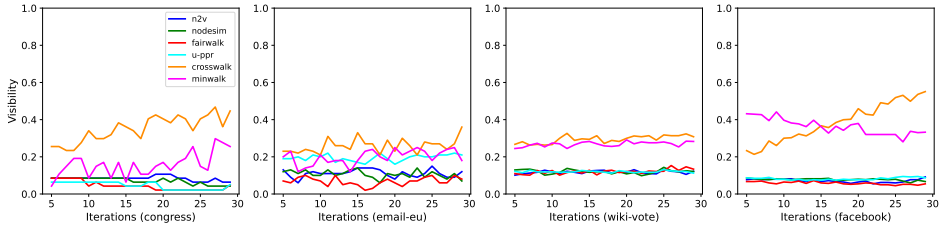


Figure 2.3: Evolution of visibility over iterations for minority size of 15%. We observe that visibility remains relatively stable for all methods, with the exception of 'crosswalk', which consistently shows an increase. In some cases, this increase is concerning, such as on 'facebook', where it grows beyond 50%. Generally, our 'minwalk' algorithm generates significantly higher visibility compared to other methods, with the notable exception of 'crosswalk'. The very high and continually increasing visibility levels produced by 'crosswalk' might be considered excessive for a minority size of 15%.

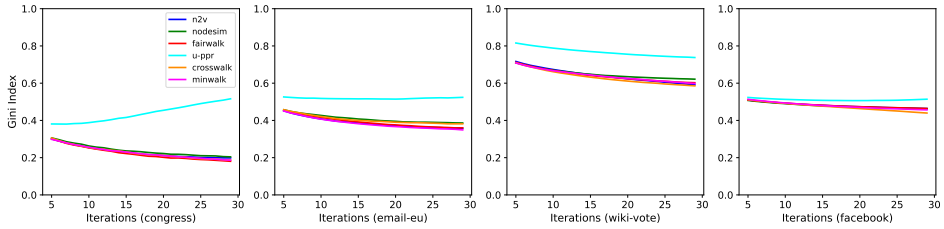


Figure 2.4: Evolution of the Gini index over iterations for minority size of 15%. All algorithms, with the exception of 'u-ppr', tend to decrease the Gini index over time. A decreasing Gini indicates a more balanced degree distribution within the network.

Results In our experiments, we focused on analyzing network dynamics changes after 30 iterations. Also, we set the length of random walks for the random-walk-based algorithms to 40. The results regarding minority visibility, clustering coefficient, and Gini coefficient are presented in Figures 2.1 and 2.2, specifically with a minority size of 15%.

Minority Visibility. Figure 2.1 helps us make the following observations related to minority visibility, particularly when the minority size is set at 15%. The first clear finding is that the 'crosswalk' algorithm significantly boosts minority visibility, often exceeding 15% and, in some cases, reaching over 50%, as observed in the 'facebook' dataset. However, we should also consider the implications of excessively amplifying minority visibility. For instance, a visibility surge beyond 50% in a 15% minority context, could be seen as counterproductive, potentially provoking a backlash from the majority. This scenario underscores the need for a balanced approach to avoid such pitfalls.

Our 'minwalk' algorithm increases the visibility of minority groups more moderately than 'crosswalk', but also significantly outperforms competing algorithms like 'n2v', 'nodesim', 'fairwalk', and 'u-ppr'. The last three algorithms, 'nodesim', 'fairwalk', and 'u-ppr', despite being designed with fairness and diversity in mind, fall short in achieving a good minority visibility. Their focus on optimizing other metrics leads to a disproportionate decrease in minority visibility, often significantly below the actual minority proportion, as evidenced in datasets such as 'congress', 'email-eu', 'wiki-vote', and 'facebook'. Moreover, these algorithms do not significantly differ in performance from the baseline 'n2v' algorithm in terms of increasing the minority visibility.

Gini Coefficient. After 30 iterations at 15% minority size, we assessed the network's degree distribution balance using the Gini coefficient, shown in Figure 2.2. Higher Gini values indicate greater imbalance. Our findings reveal that 'u-ppr' typically causes the greatest imbalance across all networks. Notably, 'nodesim' often emerges as the next most imbalancing algorithm, with the exception of Facebook. In the case of Facebook, 'nodesim' ties

for this position with ‘n2v’ and ‘fairwalk’. Our ‘minwalk’ algorithm typically results in Gini coefficients on the lower end, indicating its propensity to move the network towards a more equitable state with reduced connectivity disparities.

Evolution of Visibility and Gini Coefficient. Figures 2.3 and 2.4 illustrate the evolution of minority visibility and Gini coefficient across various datasets for a minority size of 15%.

In the evolution of visibility, we note that for most methods, visibility remains relatively constant, except for ‘crosswalk’, which shows a consistent increase over iterations. This rise is particularly notable in cases like ‘facebook’, where visibility exceeds 50%. Our ‘minwalk’ algorithm generally yields significantly higher visibility than other methods, except for ‘crosswalk’. However, the high and continually increasing visibility by ‘crosswalk’ is excessive for a minority size of 15%, potentially leading to backlash and risk of abandonment of the recommender algorithm by the majority. In datasets like Facebook, methods such as ‘n2v’, ‘nodesim’, ‘fairwalk’, and ‘u-ppr’ result in minority visibility below the 15% threshold, which is not ideal. Our goal is to achieve fair representation of minorities, approximately at their 15% proportion (or higher), within the top 10% of nodes ranked by PageRank.

Figure 2.5 presents the evolution of visibility across different minority sizes over iterations on the ‘facebook’ dataset. For algorithms such as ‘n2v’, ‘nodesim’, and ‘fairwalk’, we notice a relatively flat trend in visibility changes, generally staying below the corresponding minority percentage. A notable deviation is seen with ‘fairwalk’ at the 25% minority size, where it markedly boosts visibility beyond 25%. On the other hand, ’u-ppr’ shows impressive visibility results for the 25% minority size, but this visibility significantly drops below the respective minority percentages for other sizes. ‘Crosswalk’ displays a concerning trend, seemingly ignoring minority size and frequently increasing minority visibility past the 50% threshold. In contrast, our ‘minwalk’ algorithm initially exhibits somewhat elevated visibility levels, yet it progressively adjusts to more appropriate levels as time

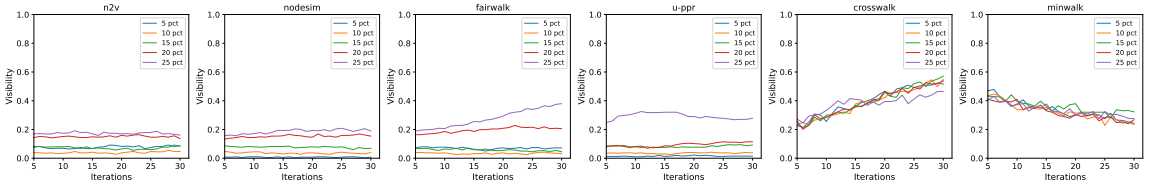


Figure 2.5: Visibility evolution over iterations for different minority sizes on the 'facebook' dataset. For 'n2v', 'nodesim', and 'fairwalk', we observe a relatively flat behavior in terms of visibility change, mostly lower than the minority percentage. 'Fairwalk' exhibits an exception at the 25% minority size, where it significantly increases visibility well beyond 25%. 'u-ppr' produces strong visibility for the 25% minority size, but for other sizes, visibility is markedly reduced below their corresponding minority percentages. 'Crosswalk' seems indifferent to the minority size, concerningly increasing the visibility of minorities beyond the 50% mark. Finally, our algorithm, 'minwalk', starts somewhat high, but over time, it normalizes the visibility to more acceptable levels.

progresses.

Crosswalk and Nodesim: Varying Parameters. Finally in Table 2.2, we study how varying the parameters of 'crosswalk' and 'nodesim' impact the visibility of the 15% minority. For both systems, altering parameters does not significantly change the results concerning minority visibility. Specifically, Crosswalk tends to increase minority visibility excessively, often exceeding the 15% mark. In contrast, NodeSim, despite parameter changes, maintains minority visibility at modest levels, typically below the 15% threshold. Similar behaviour was also observed for the clustering and gini coefficients.

α / p	1.0	2.0	3.0
0.1	0.560794	0.526055	0.491315
0.5	0.538462	0.493797	0.531017
1.0	0.508685	0.550868	0.573201

α / β	2.0	3.0	4.0
1.0	0.121588	0.086849	0.119107
2.0	0.114144	0.089330	0.104218
3.0	0.104218	0.099256	0.099256

Table 2.2: **[Left.]** Crosswalk: Visibility of 15% minority for different values of its α and p parameters. **[Right.]** Nodesim: Visibility of 15% minority for different values of its α and β parameters.

2.6 Conclusions

In this chapter, we investigated the impact of fairness- and diversity-aware link recommendation algorithms on the structural dynamics of social networks, with a particular focus on the visibility of minority communities. Our goal was to understand not only how different algorithms perform with respect to minority exposure but also how their repeated application over time influences overall network structure and equity.

Unlike previous studies that rely on synthetic datasets or attribute-based group definitions (e.g., gender or age), our work used real-world networks and grounded its fairness analysis in structural communities—groups of users identified based on link density. This approach allowed us to capture naturally occurring minority clusters that do not necessarily align with traditional demographic categories, expanding the relevance and applicability of fairness in social recommendation tasks. Structural minorities, such as isolated professional circles or niche interest communities, often face reduced visibility in standard recommendation pipelines. Addressing fairness for these groups is vital for cultivating diverse and inclusive online ecosystems.

We examined the performance of several prominent algorithms, including the classical 'node2vec', the personalized propagation approach 'u-ppr', and three fairness- or diversity-aware methods: 'fairwalk', 'nodesim', and 'crosswalk'. Our empirical results revealed important differences. 'crosswalk' stood out for its substantial gains in minority visibility, outperforming all others across multiple networks and settings. However, this visibility boost came at the cost of overexposure, with minority nodes receiving a disproportionately high share of recommendations. Such overcompensation risks disrupting user behavior and reducing engagement among majority groups, potentially leading to algorithmic backlash.

On the other end of the spectrum, 'u-ppr', 'nodesim', and 'fairwalk' failed to meaningfully increase minority visibility, with some even underperforming the baseline 'node2vec'. This highlights a critical challenge: existing fairness-aware methods may encode fairness

objectives but still fail to address the visibility needs of minority groups in dynamic network settings.

To address this gap, we introduced ‘minwalk’, a novel algorithm that provides a more balanced treatment of minority visibility. Unlike methods that aggressively promote minorities without considering network-wide effects, ‘minwalk’ incrementally enhances visibility while maintaining a stable and diverse recommendation distribution. Our experiments demonstrated that ‘minwalk’ successfully increased minority visibility across time steps, performed competitively with state-of-the-art methods, and maintained low levels of popularity bias.

Further, we analyzed the Gini coefficient of degree distribution under each algorithm to assess network-level equity. All algorithms, with the exception of ‘u-ppr’, led to a decline in the Gini coefficient, indicating reduced concentration of link recommendations on a few popular nodes. This structural outcome reflects a broader shift toward a more equitable distribution of attention across the network, which is essential for reducing centralization and promoting diversity.

Taken together, our results highlight the complexity of operationalizing fairness in social network algorithms. Optimizing for fairness is not simply a matter of equalizing exposure or promoting minority nodes in isolation. The network context, temporal feedback effects, and user retention dynamics all play critical roles. Algorithms that ignore these factors may cause unintended disruptions or fail to produce lasting structural changes.

This work made several key contributions. First, it redefines fairness in link recommendations through the lens of structural communities, moving beyond attribute-based (identity) approaches. Second, it introduces ‘minwalk’, a principled yet pragmatic algorithm that delivers minority visibility without destabilizing the network. Third, it offers a temporal evaluation framework to understand how fairness-aware algorithms evolve and interact with user behavior over time.

Several promising directions remain for future work. One path is to explore adaptive algorithms that dynamically adjust fairness objectives in response to user feedback or evolving community structures. Another is to integrate user-level satisfaction models or behavioral data to better balance engagement with fairness goals. A third direction involves examining the intersection of structural and attribute-based group definitions, to ensure that fairness interventions are inclusive across multiple axes of identity and behavior. Finally, future studies could investigate the long-term societal effects of visibility shifts in recommender systems, particularly in politically or culturally sensitive domains.

Uncovering both the benefits and risks of fairness-aware algorithms, this chapter provides a foundation for designing more equitable and sustainable social recommendation systems that support diverse user participation while preserving the integrity of the network.

Chapter 3

Word Embedding Bias in LLMs

3.1 Introduction

While the previous chapter examined how fairness-aware algorithms can reshape the visibility of minority groups within network structures, this chapter turns to a different but equally critical dimension of algorithmic bias: language. Large Language Models (LLMs), increasingly deployed in applications from chatbots to recommendation systems, rely on word embeddings—numeric representations of meaning learned from vast text corpora. These embeddings do not just capture linguistic patterns; they also absorb and encode social biases present in the data. As a result, LLMs often reflect and amplify demographic biases related to gender, race, and other social identities. These biases are not merely theoretical—they have concrete consequences in real-world systems, influencing hiring decisions, educational tools, and content moderation.

Recognizing and mitigating these biases is essential for building equitable NLP technologies. A growing body of research has demonstrated that embeddings encode societal bias, whether explicitly or implicitly [10, 13]. In a seminal study, Caliskan et al. (2017) introduced the Word Embedding Association Test (WEAT), which showed that pretrained

GloVe embeddings exhibit significant bias across dimensions such as gender, race, and age. For example, words associated with women were disproportionately linked to caregiving, while those associated with men were linked to competence and careers. Such associations risk reinforcing existing social inequalities, particularly in domains like hiring, education, and healthcare, where automated systems increasingly influence decisions.

Building on this foundation, Caliskan et al in [14] expanded the analysis by examining not just semantic associations between words, but also the frequency, syntax, and broader categories of biased words. By analyzing factors like word frequency and associations with areas such as big-tech, [14] discovered deeper gender biases not present in their previous work [13]. Additionally, they introduced the SC-WEAT method as a central tool for quantifying biases.

Our work significantly extends the scope of prior research on bias in word embeddings by addressing four key areas: (1) the frequency of gender- and race-associated words in modern large language models (LLMs); (2) bias variation across different frequency ranges and effect sizes in these models; (3) the identification and clustering of gender- and race-associated thematic concepts; and (4) the manifestation of biases in the tech industry and higher education. Unlike Caliskan et al. in [14], which focused on older embedding models like GloVe and FastText and examined only gender bias, we expand our analysis to include modern LLM embeddings and race-associated biases. Moreover, while Caliskan et al. provided a limited conceptual analysis, we offer a more detailed examination of the thematic clusters related to both gender and race.

More specifically, we analyzed gender and race biases in the 100,000 most frequent words from the GloVe dataset, focusing on five modern contextual word embedding models– OpenAI, Cohere, Google, Microsoft E5, and BGE. Using the SC-WEAT test, we quantified biases across various frequency ranges (top 100, 1,000, 10,000, and 100,000 words), measuring word associations with specific attributes to determine the direction and magnitude

of bias. Additionally, we used k-means clustering and t-Distributed Stochastic Neighbor Embedding (T-SNE) visualizations to identify the semantic categories of gender- and race-associated words. We used a bottom-up approach to cluster 10,000 word associations and utilized GPT-3.5 to visualize key bias concepts in LLMs. We further analyzed the cosine similarity of words associated with Big Tech and top universities, identifying the top 1,000 word associations for each attribute. Our analysis provides critical insights into how biases in word embeddings are reflected in real-world contexts, particularly in the tech industry and higher education. Our study takes an important step in revealing hidden biases within popular large language models, moving toward understanding their impact on model behavior.

3.2 Related Work

3.2.1 Word Embedding Algorithms

Word embeddings, which map words to numeric vectors, have advanced natural language processing (NLP) by capturing semantic relationships within text. While GloVe [70] and similar models like Word2Vec [64] laid the groundwork for word embedding, modern LLM-based embeddings have since emerged, such as those from OpenAI, Cohere, Google, Microsoft (E5), and the Beijing Academy of Artificial Intelligence (BGE-M3) (among others). These models adapt word representations based on their surrounding context, providing a more nuanced understanding of language [31]. Despite their sophistication, these embeddings still exhibit cultural stereotypes and biases, which remain challenging to mitigate [8]. Our work extends beyond [14] to explore these modern LLM embedding frameworks.

3.2.2 Measuring Bias in Word Embeddings

The Word Embedding Association Test (WEAT), introduced by Caliskan et al. [13], quantifies biases in word embeddings by measuring the differential association of two sets of target words with two sets of attribute words. For instance, it has shown that terms like ”engineer” and ”scientist” are often associated with male attributes, while ”nurse” and ”teacher” are associated with female attributes. Extensions of WEAT have been applied to detect various demographic biases, including gender, race, and age [18, 49].

Our study leverages the Single-Category Word Embedding Association Test (SC-WEAT) [14] to investigate biases related to gender and race. SC-WEAT, an extension of the Word Embedding Association Test (WEAT) [13, 83], quantifies the bias of a single target word relative to two sets of attribute words. The test computes an effect size (ES), using the formula:

$$ES(\vec{w}, A, B) = \frac{\text{mean}_{a \in A} \cos(\vec{w}, \vec{a}) - \text{mean}_{b \in B} \cos(\vec{w}, \vec{b})}{\text{std\_dev}_{x \in A \cup B} \cos(\vec{w}, \vec{x})}$$

Here, $\vec{w}$ is the target word, and A and B are two sets of words, each containing at least eight terms [14]. The formula calculates the mean cosine similarity between the target word and the terms in each set, normalized by the overall standard deviation. A higher positive effect size indicates a stronger association with set A (e.g., female-association), while a negative effect size indicates a stronger association with set B (e.g., male-association).

The output is an effect size, expressed as Cohen’s d [29], along with a p-value. The effect size indicates the strength of the association, and the p-value tests its statistical significance [14]. Cohen’s benchmarks define effect sizes of 0.2 as small, 0.5 as medium, and 0.8 as large [29]. For example, if a target word like “nurse” shows a large positive effect size (e.g., 0.8) with the female-association set, it suggests a strong gender bias associating nursing with women. Conversely, a negative effect size (e.g., -0.8) would imply a stronger

male-association.

To evaluate the significance of the observed effect size, a permutation test is applied. In this process, the associations between sets A and B are shuffled by randomly reassigning words to each set, effectively mixing their labels. The mean difference in associations is recalculated for each shuffle, and this procedure is repeated 10,000 times. The resulting distribution of mean differences represents the effect size expected under random conditions. The p-value is obtained by comparing the observed effect size to the distribution of randomly generated effect sizes.

Caliskan et al. [14] used SC-WEAT to reveal entrenched gender stereotypes in widely used word embeddings such as GloVe and FastText.

3.2.3 How is the present work different?

In this work, we extend the bias analysis of [14] to five contemporary large language model-based word embedding models: OpenAI, Cohere, Google, Microsoft, and BGE. We go beyond gender analysis to include race bias, addressing four key dimensions: the frequency of gender- and race-associated words, variations in biases across different frequency ranges and effect sizes, the clustering of related thematic concepts, and how these biases manifest in the tech industry and higher education.

3.3 Data

Most Frequent Words. We focus on the most frequent words from the GloVe embedding dataset, which contains 2.2 million words [70]. After filtering, we select the top 100,000 words for our analysis. We emphasize that, unlike [14], we use GloVe solely to identify the most frequent words, without using the GloVe embeddings themselves, as our focus is on the aforementioned LLM embeddings.

Word Embedding Models. We use five contextual embedding models: OpenAI's text-embedding-3-small (1,536 dimensions) [69], Microsoft's E5-large-v2 (1,024 dimensions) [88], Google's text-embedding-004 (300 dimensions) [27], Cohere's embed-english-v3.0 (1,024 dimensions) [30], and BGE's BGE-M3 (1,024 dimensions) [20]. Each model provides high-dimensional embeddings for semantic analysis.

Stimuli Words (Attribute Sets). Gender bias is measured using gender stimuli from Caliskan et al. [13], where positive effect sizes indicate a female association and negative sizes indicate male. Race bias uses ChatGPT-3.5-generated stimuli to measure biases among White, Asian, and Black groups. Positive effect sizes indicate a White or Asian association, depending on the comparison [83]. We show our stimuli words in Table 3.1.

Table 3.1: Gender and Race Stimuli

Category	Stimuli Group	Stimuli Words
Gender	Female	Female, Woman, Girl, Hers, Sister, She, Her, Daughter
	Male	Male, Man, Boy, Brother, He, Him, His, Son
Race	White	American, Australian, British, Canadian, White, Caucasian, European, French, German, Italian
	Asian	Asian, Chinese, Japanese, Indonesian, Indian, Korean, Pakistani, Thai, Filipino, Brown
	Black	African, African-American, Black, Congolese, Egyptian, Ethiopian, Haitian, Jamaican, Kenyan, Nigerian

Big Tech Words. We select Big Tech companies based on [1] that are present in the top 100,000 GloVe words, including companies such as Google, Amazon, Facebook, and Microsoft.

Top University Words. The top 50 universities from the 2024 Times Higher Education rankings [36] are added to the word list after normalizing their names.

3.4 Approach

Most Frequent Words Extraction. After we obtain the most frequent words, we apply the following preprocessing steps to clean up the data:

1. Remove stopwords, punctuation words with non-English characters, digits, and exclude words containing any of these.

2. Filter out words with fewer than three characters long.

3. Include the stimuli words used in SC-WEAT analysis to ensure accuracy benchmarks (See Table 3.1).

This cleaned set of frequent words is then input into five embedding models: OpenAI, Cohere, Google, E5 Microsoft, and BGE (BAAI General Embedding), generating the embeddings used for further analysis.

Frequency of Gender- and Race-Associated Words. We apply SC-WEAT to observe gender and race biases in the top 100, 1,000, 10,000, and 100,000 most frequent words generated from each embedding model. For gender bias, we analyze associations between female and male groups, assigning positive effect sizes to female-associated words and negative effect sizes to male-associated words. For race bias, we analyze three groups: White, Asian, and Black. Pairwise comparisons (White vs. Black, White vs. Asian, and Asian vs. Black) are performed, with positive effect sizes indicating associations with Whites (or Asians in the third comparison), and negative values indicating associations with Blacks or Asians, as relevant.

Bias Analysis by Frequency Range and Effect Size. We quantify gender and race biases across different frequency ranges by computing bias strength using SC-WEAT. Bias strength is evaluated based on effect size thresholds, following Cohen's classification [29]: Null bias: 0.00 – 0.19 Small bias: 0.20 – 0.49 Medium bias: 0.50 – 0.79 Large bias: ≥ 0.80.

We apply these thresholds to the top 100, 1,000, 10,000, and 100,000 most frequent words extracted from each embedding model. A higher effect size indicates stronger bias, with greater disparities in word associations between the groups (e.g., gender or race). Models exhibiting larger effect sizes are considered to have more pronounced biases, as they display significant differences in the word distributions between demographic groups. This classification allows for a clear comparison of bias levels across embedding models and frequency ranges.

3.4.1 Semantic Categories of Gender- and Race-Associated Words

To better understand the nature of gender and race biases, we identify strong associations between demographic groups and specific stereotypes. Using SC-WEAT, we categorize words into two groups based on effect size and p-value:

- **Group 1**: The 1,000 most frequent words with an effect size ≥ 0.50 and a p-value < 0.05, indicating strong positive associations.
- **Group 2**: The 1,000 most frequent words with an effect size ≤ -0.50 and a p-value < 0.05, indicating strong negative associations.

In the gender bias analysis, Group 1 represents words associated with female attributes, while Group 2 represents words associated with male attributes. For race bias, Group 1 includes words associated with the White attribute group (in the White vs. Black and White vs. Asian comparisons) and the Asian attribute group (in the Asian vs. Black comparison). Group 2 represents words associated with Black individuals (in the White vs. Black and Asian vs. Black comparisons) and with Asians (in the White vs. Asian comparison).

To further explore patterns, we applied K-means clustering (using the Elkan algorithm) to these two groups of 1,000 words each. The optimal number of clusters, determined using

the elbow method, is $k = 11$. We then reduced the dimensionality of the clustered embeddings using t-SNE for 2D visualization. Finally, we employed ChatGPT 3.5 to analyze and assign thematic concepts to each cluster, revealing common patterns and stereotypes linked to the identified biases.

3.4.2 Bias in Big Tech and Higher Education Contexts

Big Tech Bias Analysis

We examine the representation of bias within Big Tech by focusing on the common Big Tech words identified by Abdalla and Abdalla [1]. These include Google, Amazon, Facebook, Microsoft, Apple, Nvidia, Intel, IBM, Huawei, Samsung, Uber, and Alibaba. From the 100,000 most frequent words, we calculate the cosine similarity of the embeddings for these Big Tech terms and identify the top 10,000 most associated words. For consistency, we identify the most associated words by intersecting the top 10,000 most associated words from all five models, resulting in a consistent set of 622 Big Tech-associated words. We then apply SC-WEAT to this 622-word set, using effect size ranges of 0.00 – 0.19 (null), 0.20 – 0.49 (small), 0.50 – 0.79 (medium), and ≥ 0.80 (large) to observe the bias strength for each class in pairwise comparisons for each of the five embedding models we consider.

Higher Education Bias Analysis

For the higher education context, we compile a list of the Top 50 universities from the Times Higher Education 2024 ranking [36]. Since these universities are absent from our frequent word list, we extract their embeddings and append them to the word set. We then compute the cosine similarity with these university embeddings, identifying the top 10,000 associated words. Intersecting these word sets from all five models produces a consistent set of 1,120 Higher Education-associated words. We then apply SC-WEAT to this set, using effect size ranges (0.00 – 0.19, 0.20 – 0.49, 0.50 – 0.79, ≥ 0.80) to observe bias

strength in pairwise comparisons for each of the five embedding models we consider.

3.5 Results

3.5.1 Top-Word Association Bias

Gender Association of Top-k Words. Figure 3.1 presents the distribution of gender-associated words across five different word embedding models (BGE, OpenAI, Cohere, Google, and Microsoft) for various sizes of word sets (top 100, 1,000, 10,000, and 100,000 words). The chart uses blue to indicate the percentage of male-associated words and pink for female-associated words.

Across most models, there is a consistent trend of greater male association, which tends to decrease as the size of the word sets increases. For instance, in the BGE Model, male-associated words account for 86% at the top 100 words but decrease to 54% at the top 100,000 words. Similarly, the Cohere Model shows a decrease from 79% male association at the top 100 words to 67% at the top 100,000 words. The observed decrease in male-associated word bias as the size of the top word sets increases indicates that models become less biased with a broader selection of frequently used words.

Interestingly, the OpenAI Model exhibits a different trend. Here, female-associated words dominate, ranging from 62% at the top 100 words to 63% at the top 100,000 words. This pattern indicates a reversal of the common trend seen in the other models, suggesting a unique alignment in the OpenAI model toward female associations across all analyzed word set sizes.

The Google and Microsoft models also show a consistent male bias, though with less variation across different word sets. For example, the Microsoft Model maintains a high percentage of male-associated words, from 83% at the top 100 words to 86% at the top 100,000 words, indicating strong and persistent male bias regardless of the word set size.

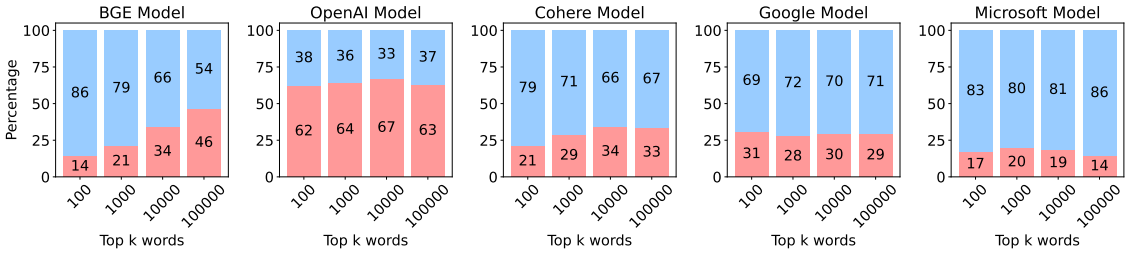


Figure 3.1: Gender Association of Top Words. Male is light blue, female is pink.

Race (White vs Black) Association of Top-k Words. Figure 3.2 (1st row) presents the association distribution of the most frequent words between White and Black attribute sets. As before, the analysis spans different word set sizes, focusing on the top 100, 1,000, 10,000, and 100,000 words. For instance, in the BGE model, 95% of the top 100,000 words are associated with White attributes, while only 5% are associated with Black attributes. This strong skew toward White-associated words remains consistent across different word set sizes. Notably, all the models exhibit a strong association bias toward White.

Interestingly, the Google model is the most balanced among all the models, with 24%, 12%, 12%, and 18% Black association for the top 100, 1,000, 10,000, and 100,000 words, respectively. This distribution shows a noticeable reduction in bias compared to the other models. Still, even for the Google model, the association bias towards White remains. The OpenAI model is next in terms of balance, exhibiting a higher Black association of top words compared to the other models (except Google, which performs the best) but still showing a significantly skewed distribution towards White-associated words in the larger sets. Such an imbalance of association toward White across all the models highlights the persistent issue of racial bias, despite varying degrees of mitigation efforts.

Race (White vs. Asian) Association of Top-k Words. Figure 3.2 (2nd row) shows the percentage distributions of race association of top words (White vs. Asian) across the five embedding models. The BGE and OpenAI models maintain a relatively balanced

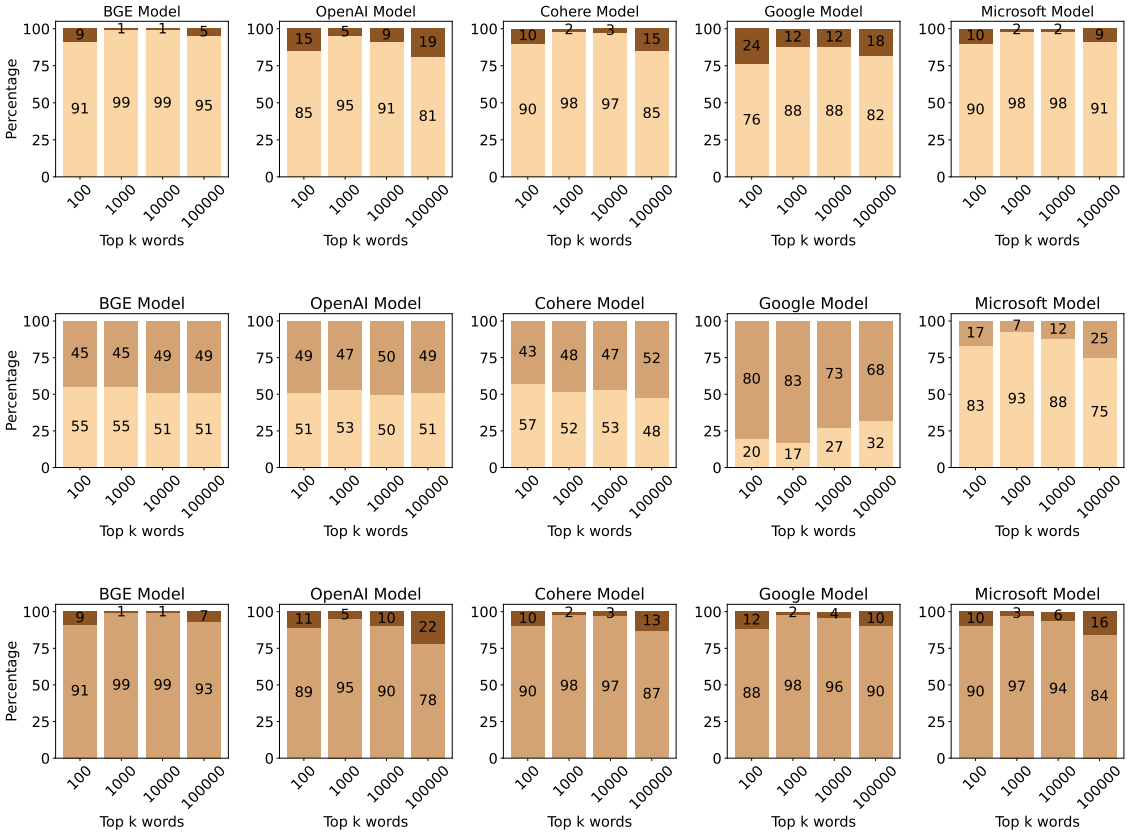


Figure 3.2: Race Association of Top Words. 1st row: White (lighter color) vs Black (darker color); 2nd row: White (lighter color) vs Asian (darker color); 3rd row: Asian (lighter color) vs Black (darker color).

distribution between White and Asian associations. The Cohere model is also balanced, albeit less so than the previous two models. Surprisingly, the Google and Microsoft models show the most bias. The Google model is heavily skewed toward Asian associations, while the Microsoft model favours White associations.

Race (Asian vs. Black) Association of Top-k Words. Figure 3.2 (3rd row) shows the percentage distributions of race associations for top words (Asian vs. Black) across the five models. All models show a strong skew toward Asian associations, with Black associations being consistently underrepresented. Among them, the BGE model exhibits the strongest bias toward Asian associations, whereas the OpenAI model displays the least. These find-

ings highlight a noticeable disparity in how the models represent Asian and Black groups.

3.5.2 Top-Word Association by Effect Size.

Table 3.2 presents the effect size analysis of gender associations for the top 100, 1,000, 10,000, and 100,000 words using SC-WEAT across the five models, with effect sizes ranging from 0 to 0.8, indicating the strength of gender association. Similarly, Table 3.3 provides the effect size analysis for race associations.

Across most models, there is a clear trend where male-association of top words consistently outnumbers female-association across all effect sizes. Notably, OpenAI deviates from this pattern, showing a higher number of female-associations across all top-k sets and effect size levels. For instance, within the top 100,000 words, the OpenAI model reports the highest number of strong female-associations (+0.8), with 20,174 words. In contrast, the Google and Microsoft models show the highest number of strong male-associations for the top words.

For White vs. Black associations, the BGE model has the highest number of strong White-associations (+0.8), with 52,707 words, while OpenAI shows the highest number of strongly Black-association words (-0.8), with 3,028 words, although Black associations remain underrepresented in all models. For White vs. Asian associations, OpenAI has the highest number of strong White-association words (+0.8), with 7,961 words, while Google shows the most strong Asian-association words (-0.8), with 19,465 words. Lastly, for Asian vs. Black associations, BGE has the most strong Asian-association words (+0.8), with 57,440 words, while OpenAI has the highest number of strong Black-associations (-0.8), with 3,404 words, yet Black associations remain consistently underrepresented across all models (See also Section 3.7 for more details).

Table 3.2: Gender-Associations by Effect Size: number of top-100,000 words associated with female and male attributes. The 0, 0.2, 0.5, 0.8 columns denote the number of words with an effect size between 0 and 0.2, 0.2 and 0.5, and so on.

LLM	Female				Male			
	0	0.2	0.5	0.8	0	0.2	0.5	0.8
BGE	53,631	42,980	28,055	15,825	46,369	35,333	20,293	8,998
OpenAI	62,902	51,678	34,871	20,174	37,098	27,198	14,912	6,629
Cohere	33,160	24,315	14,378	8,604	66,840	56,283	39,141	23,695
Google	29,288	22,778	14,951	9,275	70,712	63,081	49,655	34,546
Microsoft	14,345	8,553	4,705	2,895	85,655	75,282	50,949	26,358

Table 3.3: Race Associations by Effect Size (Top 100,000 words)

LLM	White vs. Black							
	White				Black			
	0	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
BGE	95,832	91,344	77,736	52,707	4,168	1,879	544	134
OpenAI	80,654	72,159	54,904	33,649	19,346	12,889	6,604	3,028
Cohere	85,446	77,539	60,416	36,263	14,554	9,132	4,048	1,493
Google	82,104	73,213	54,736	32,319	17,896	11,228	5,133	2,010
Microsoft	91,012	83,785	63,970	34,996	8,988	4,834	1,787	556
LLM	**White vs. Asian**							
	White				**Asian**			
	0	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
BGE	51,881	31,694	11,091	2,600	48,119	28,248	9,534	2,241
OpenAI	50,882	38,086	20,833	7,961	49,118	36,798	20,833	9,147
Cohere	48,724	32,613	13,457	3,624	51,276	35,780	17,645	7,109
Google	32,263	22,406	11,439	4,485	67,737	56,365	37,510	19,465
Microsoft	74,870	59,300	29,510	7,497	25,130	15,330	7,205	3,178
LLM	**Asian vs. Black**							
	Asian				**Black**			
	0	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
BGE	92,803	87,553	75,710	57,440	7,197	3,862	1,171	270
OpenAI	78,270	69,567	53,272	34,445	21,730	15,070	7,779	3,404
Cohere	86,816	79,657	64,012	40,730	13,184	8,338	3,755	1,300
Google	90,015	83,964	69,988	48,838	9,985	6,088	2,550	928
Microsoft	83,823	71,859	46,359	19,632	16,177	8,562	2,685	650

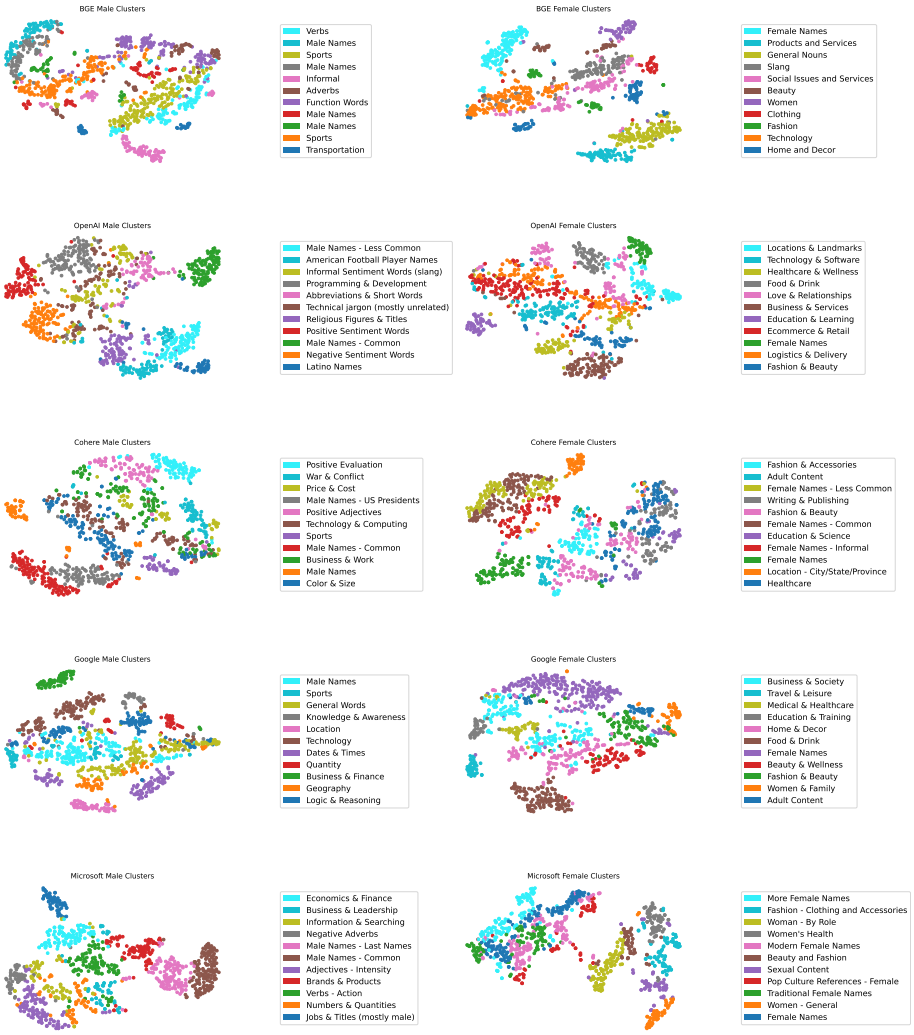


Figure 3.3: Clusters for gender

3.5.3 Semantic Categories of Gender and Race Associated Words.

For each attribute in the gender and race analysis, we identified eleven clusters from each set of the 1,000 most frequently used female-, male-, White-, Asian-, and Black-biased words, each with an effect size greater or equal to 0.50 and a p-value less than 0.05. We then input these cluster groups into GPT-3.5 to label each cluster, obtaining the following results.

Gender. We identified five female-associated cluster sets from each gender analysis across five different models (see Figure 3.3). Common female-associated clusters are related to healthcare, home decor, beauty, fashion, and sexual content. Additionally, female names are consistently classified as a common theme. Conversely, male-associated clusters commonly focus on technology, sports, business, and sentiment words, with male names classified as a theme across all models. Each gender cluster set includes some noise in the form of generic titles, such as "name", "miscellaneous," "verb," or "adjective," which does not provide clear and meaningful categorization.

Race. We identified ten White-associated, ten Asian-associated, and ten Black-associated cluster sets from pairwise race analyses across five different models. Common White-associated clusters include business, people & society, education, media, and technology. In contrast, Asian-associated clusters frequently relate to business, software engineering, technology, entertainment, and food and culture. Black-associated clusters typically focus on religion, music, athletes and public figures, wild animals, and ethnicity. Each race cluster set includes some noise in the form of generic titles, such as "name", "location", "noun", "adverb", or "adjective", which does not provide clear and meaningful categorization. The figures for these clusters can be found in our github repository `github.com/Poomon001/Bias-in-Word-Embeddings`.

3.5.4 Gender and Race Bias in Big Tech Industry.

Gender. Our results indicate that 3 out of 5 models show a stronger association between big tech words and males. In the Cohere, Google, and Microsoft models, over 50% of the total 622 big tech words are associated with men at an effect size of 0.5, while fewer than 10% are associated with women (see Figure 3.4, top). In contrast, the OpenAI model reports a significant association between big tech words and women, while the BGE model

indicates minimal gender bias in the tech field.

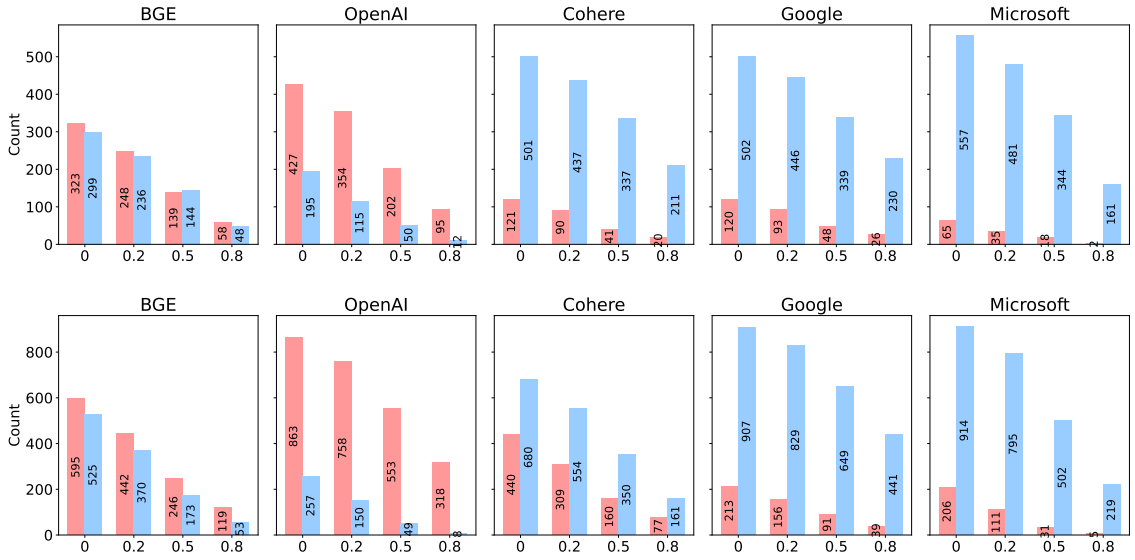


Figure 3.4: Big-Tech and Top-University Gender-Association (Female vs Male). Effect sizes on the x-axis. 1st row: Big-Tech, 2nd row: Top-University.

Race. In pairwise comparisons, 4 out of 5 models show that big tech words are primarily associated with Asians rather than Whites (See Figure 3.5). The exception is the OpenAI model, which slightly favours Whites over Asians. All five models indicate a significant association of big tech words with Asians and Whites compared to Blacks. Notably, Black attributes have a minimal association with big tech.

3.5.5 Gender and Race Bias in Higher Education.

Gender. Our results indicate that 3 out of 5 models show a stronger association between top university words and males. This is similar to the results we observed for Big Tech words. In the Cohere, Google, and Microsoft models, over 30%, 50%, and 40%, respectively, of the 1,120 top university words are associated with men at an effect size of 0.5, while only 5% or less are associated with women. In contrast, the BGE and OpenAI models report a higher association between top university words and women rather than men.

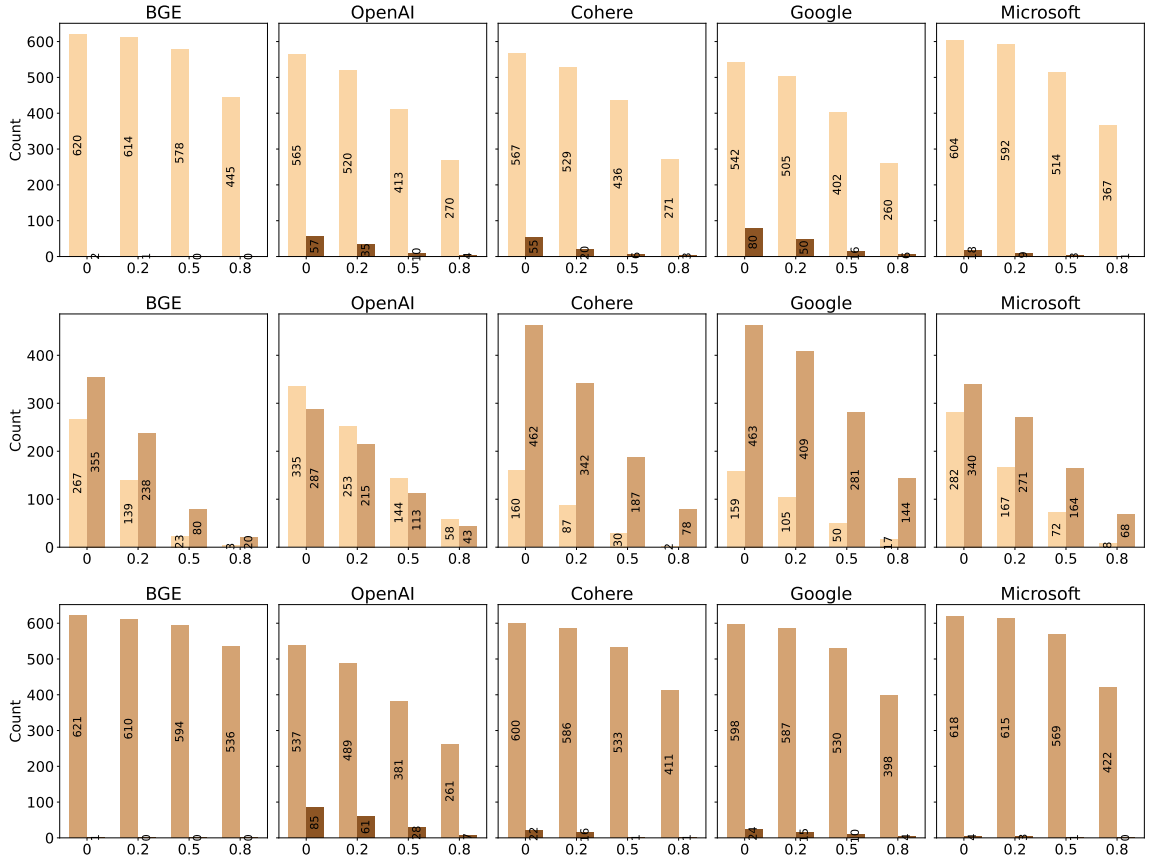


Figure 3.5: Big-Tech Race Association. Effect sizes on the x-axis. 1st row: White (lighter color) vs Black (darker color); 2nd row: White (lighter color) vs Asian (darker color); 3rd row: Asian (lighter color) vs Black (darker color).

Race. In pairwise comparisons, 3 out of 5 models reveal a stronger association between top university words and Whites rather than Asians. The remaining two models prefer Asians over Whites. Across all five models, top university words are significantly more associated with Asians and Whites rather than Black attributes, which show minimal association with these words. We show the charts in Section 3.7.

3.6 Conclusions

Our study reveals several surprising patterns of gender and race bias across modern large language models, exposing clear disparities that extend beyond what previous research

has shown. For instance, the analysis of gender association highlights that, unlike most models, the OpenAI model exhibits a reversal of the common male bias trend, showing a higher proportion of female associations across all word set sizes. This unique pattern contrasts sharply with the consistent male association found in other models.

Similarly, when examining race associations, we discovered that Black attributes are strikingly underrepresented across all models, and the few strong Black associations typically involve specific domains like public figures and athletes. In contrast, Asian associations dominate in many models, particularly in the BGE model, which shows an overwhelming skew toward Asian associations across multiple word set sizes.

Our study highlights the pervasive nature of gender and race biases in modern large language models, revealing surprising patterns across multiple dimensions. Our analysis uncovered several unexpected results, such as the distinct ways biases manifest across gender and race, as well as differences between models. These findings underscore the need for ongoing research and more robust debiasing strategies by the companies providing LLMs in order to promote fair and inclusive AI systems. Addressing these issues is crucial as LLMs are increasingly used in various high-impact applications.

3.7 Additional Results for LLM Comparisons

In this section, we present additional experimental results that were omitted from the main experiments section to maintain clarity and conciseness in our discussion.

Table 3.4: Gender-Associated Words by Effect Size Using SC-WEAT: The table shows the number of words in the top-k sets (100, 1,000, 10,000, 100,000) associated with female and male attributes. Effect sizes (0, 0.2, 0.5, 0.8) denote number of words with an effect size between 0 and 0.2, 0.2 and 0.5, and so on.

	Female				**Male**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
BGE								
100	14	11	8	8	86	80	60	30
1,000	213	145	66	30	787	667	444	159
10,000	3,972	2,904	1,614	815	6,028	4,811	2,813	1,151
100,000	53,631	42,980	28,055	15,825	46,369	35,333	20,293	8,998
OpenAI								
100	62	39	23	8	38	27	15	4
1,000	643	476	250	104	357	242	102	38
10,000	6,691	5,527	3,646	1,980	3,309	2,316	1,142	454
100,000	62,902	51,678	34,871	20,174	37,098	27,198	14,912	6,629
Cohere								
100	21	16	10	10	79	66	28	16
1,000	290	177	86	36	710	569	289	112
10,000	3,427	2,325	1,148	539	6,573	5,240	3,144	1,545
100,000	33,160	24,315	14,378	8,604	66,840	56,283	39,141	23,695
Google								
100	31	22	12	8	69	62	51	37
1,000	276	182	102	49	724	633	466	307
10,000	2,956	2,188	1,271	671	7,044	6,140	4,582	2,896
100,000	29,288	22,778	14,951	9,275	70,712	63,081	49,655	34,546
Microsoft								
100	17	10	8	4	83	66	28	3
1,000	198	87	36	11	802	599	276	86
10,000	1,863	990	396	175	8,137	6,536	3,471	1,332
100,000	14,345	8,553	4,705	2,895	85,655	75,282	50,949	26,358

Table 3.5: Race-Associated Comparisons by Effect Size for BGE

BGE								
	White				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	91	90	89	83	9	8	4	0
1,000	990	989	982	941	10	8	4	0
10,000	9,938	9,837	9,455	7,963	62	36	15	1
100,000	95,832	91,344	77,736	52,707	4,168	1,879	544	134
	Asian				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	91	90	88	86	9	7	3	0
1,000	990	985	973	947	10	8	3	0
10,000	9,868	9,715	9,299	8,310	132	59	16	5
100,000	92,803	87,553	75,710	57,440	7,197	3,862	1,171	270
	White				**Asian**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	55	26	2	0	45	20	11	0
1,000	555	210	24	1	445	139	22	0
10,000	5,122	2,519	587	94	4,878	2,293	487	74
100,000	51,881	31,694	11,091	2,600	48,119	28,248	9,534	2,241

Table 3.6: Race-Associated Comparisons by Effect Size for OpenAI

OpenAI								
	White				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	85	82	70	49	15	12	9	9
1,000	959	928	785	538	41	26	11	9
10,000	9,158	8,587	7,007	4,688	842	482	177	67
100,000	80,654	72,159	54,904	33,649	19,346	12,889	6,604	3,028
	Asian				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	89	86	68	54	11	11	9	5
1,000	950	897	730	540	50	32	17	7
10,000	9,062	8,430	6,858	4,743	938	563	236	100
100,000	78,270	69,567	53,272	34,445	21,730	15,070	7,779	3,404
	White				**Asian**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	51	36	16	2	49	44	27	9
1,000	534	372	168	44	466	323	135	46
10,000	4,919	3,471	1,637	514	5,081	3,654	1,769	645
100,000	50,882	38,086	20,833	7,961	49,118	36,798	20,833	9,147

Table 3.7: Race-Associated Comparisons by Effect Size for Cohere

Cohere								
	White				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	90	86	83	64	10	10	8	3
1,000	985	976	928	709	15	14	9	3
10,000	9,746	9,431	8,383	5,824	254	119	36	12
100,000	85,446	77,539	60,416	36,263	14,554	9,132	4,048	1,493
	Asian				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	90	89	85	67	10	10	7	2
1,000	984	980	951	752	16	13	8	2
10,000	9,745	9,440	8,498	5,941	255	114	40	9
100,000	86,816	79,657	64,012	40,730	13,184	8,338	3,755	1,300
	White				**Asian**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	57	31	8	1	43	25	13	6
1,000	524	258	56	6	476	211	46	13
10,000	5,385	3,274	1,060	202	4,615	2,643	870	212
100,000	48,724	32,613	13,457	3,624	51,276	35,780	17,645	7,109

Table 3.8: Race-Associated Comparisons by Effect Size for Google

Google								
	White				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	76	66	57	36	24	17	13	8
1,000	886	815	631	395	114	64	25	11
10,000	8,814	8,039	6,236	3,846	1,186	666	224	72
100,000	82,104	73,213	54,736	32,319	17,896	11,228	5,133	2,010
	Asian				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	88	86	80	65	12	11	10	7
1,000	980	958	890	670	20	15	12	8
10,000	9,598	9,188	8,043	5,859	402	210	71	31
100,000	90,015	83,964	69,988	48,838	9,985	6,088	2,550	928
	White				**Asian**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	20	13	9	2	80	72	52	38
1,000	178	109	41	8	822	724	504	267
10,000	2,712	1,746	749	238	7,288	6,091	4,093	2,095
100,000	32,263	22,406	11,439	4,485	67,737	56,365	37,510	19,465

Table 3.9: Race-Associated Comparisons by Effect Size for Microsoft

Microsoft								
	White				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	90	89	84	67	10	10	7	1
1,000	988	984	937	703	12	11	7	1
10,000	9,858	9,655	8,668	5,887	142	73	23	2
100,000	91,012	83,785	63,970	34,996	8,988	4,834	1,787	556
	Asian				**Black**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	90	86	68	23	10	10	4	1
1,000	967	923	666	195	33	17	6	1
10,000	9,476	8,646	5,875	2,214	524	172	36	4
100,000	83,823	71,859	46,359	19,632	16,177	8,562	2,685	650
	White				**Asian**			
Top k	**0**	**0.2**	**0.5**	**0.8**	**0**	**0.2**	**0.5**	**0.8**
100	83	75	48	9	17	16	11	4
1,000	932	832	461	113	68	37	16	6
10,000	8,789	7,532	4,098	981	1,211	633	236	78
100,000	74,870	59,300	29,510	7,497	25,130	15,330	7,205	3,178

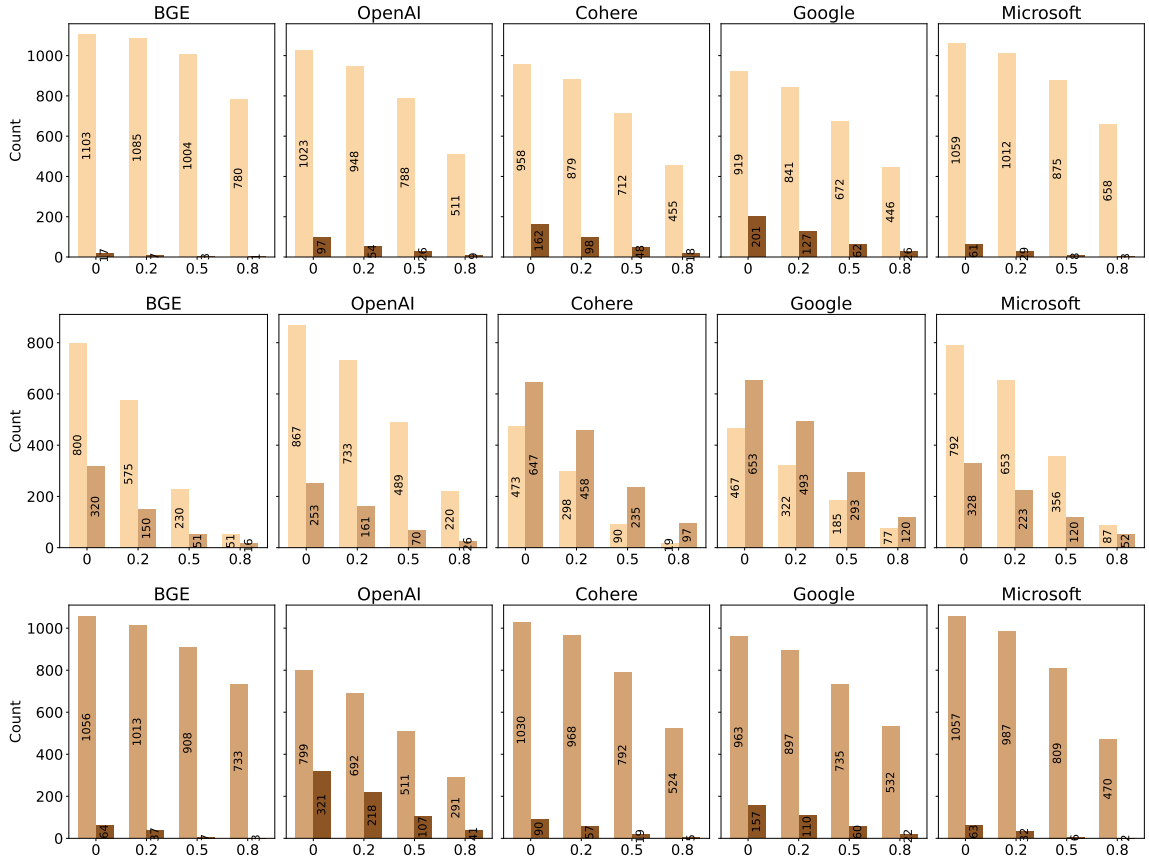


Figure 3.6: Top-University Race Association. Effect sizes on the x-axis. 1st row: White (lighter color) vs Black (darker color); 2nd row: White (lighter color) vs Asian (darker color); 3rd row: Asian (lighter color) vs Black (darker color).

Chapter 4

LLM Product Recommendation Bias

4.1 Introduction

Following our examination of how LLMs encode demographic bias in language representations, we now turn to the ways such biases manifest in practical applications—specifically, in consumer product recommendation systems. These systems, increasingly powered by LLMs, generate personalized and context-aware suggestions that influence user preferences and shape purchasing behavior across digital platforms. While effective in tailoring content, they often reflect patterns in training data that reinforce stereotypes or exclude underrepresented groups. For example, certain products may be disproportionately suggested to specific demographic groups, while others are systematically omitted, perpetuating inequality through seemingly neutral outputs.

Understanding these implicit patterns is essential for building AI systems that align with fairness and inclusion. In this chapter, we investigate how LLMs generate biased product recommendations across gender and race categories. We present a framework for detecting and analyzing these disparities, and offer insights into how such biases can be measured, interpreted, and addressed in real-world settings.

Despite progress in understanding biases in AI, prior research has predominantly focused on explicit stereotypes, leaving implicit biases–those embedded in subtle linguistic and contextual distinctions–underexplored. Cheng et al. [22] introduced the "Marked Personas" framework to measure stereotypes in LLMs using natural language prompts. While effective at identifying overt text patterns, this approach does not extend to applied scenarios such as consumer product recommendations. Similarly, Monroe et al. [67] proposed the "Fightin' Words" method to analyze group-specific language features in political discourse, but its application to consumer-focused contexts remains unexplored. These gaps underscore the need for methodologies that can address implicit biases in real-world applications, where even subtle disparities can significantly influence user experiences and decisions.

This chapter addresses these limitations by investigating implicit biases in LLM-generated consumer product recommendations. Leveraging prompt engineering, we guide the model to generate demographic-specific recommendations, enabling a detailed examination of linguistic patterns and product categories associated with different groups. By focusing on implicit biases, we reveal how LLMs can unintentionally perpetuate stereotypes, even in ostensibly neutral contexts such as product recommendations.

To systematically identify and quantify these biases, we employ three computational methods:

1. **Marked Words:** Detecting words that statistically differentiate marked groups (e.g., non-male or non-white) from unmarked groups.

2. **Support Vector Machines (SVMs):** Classifying text data to highlight distinguishing linguistic features for demographic groups.

3. **Jensen-Shannon Divergence (JSD):** Measuring disparities in word frequency distributions to capture subtle linguistic differences.

Through this multi-method approach, our analysis reveals significant linguistic and categorical disparities in LLM-generated recommendations, providing actionable insights into the biases embedded in these systems. While we do not explicitly analyze word or product embeddings in this chapter, it is important to note that LLMs rely on internal embedding representations to model concepts, generate language, and make associations. Thus, even though our methods focus on surface-level outputs, these outputs are ultimately shaped by the model's internal representations—highlighting the continued relevance of embedding-level bias in this context.

The contributions of this part are threefold. First, we introduce and formalize the problem of implicit gender and race biases in LLM-generated consumer product recommendations—a space where subtle disparities can have broad social impact. Second, we propose an integrated methodology that combines prompt engineering with computational techniques such as marked word analysis, support vector machines, and distributional divergence to detect and quantify these biases. Third, we provide empirical insights that inform the design of fairer, more inclusive recommendation systems. Together, these contributions advance the broader goal of developing AI systems that are equitable, trustworthy, and socially aware.

4.2 Related Work

Examining biases in Large Language Models (LLMs) is important as they are integrated into various applications. LLMs, trained on vast datasets, often reflect and amplify biases, perpetuating stereotypes [11, 14, 93]. This issue is especially concerning in product recommendation systems, where biased suggestions can influence purchases and reinforce disparities.

Bias Detection in LLMs. Several studies have advanced efforts to identify and reduce bi-

ases in LLMs. Cheng et al. [22] proposed the "Marked Personas" framework, using prompt engineering to reveal demographic-specific stereotypes, though it does not consider implicit biases. Caliskan et al. [13] and Chuthamsatid et al. [25] (previous chapter) showed that word embeddings reflect societal biases, highlighting the need for mitigation methods. Monroe et al. [67] introduced the "Fightin' Words" method, using log-odds ratios to identify group-specific language features. Originally designed for political discourse, this technique has been adapted to other domains but has yet to be applied to LLM-generated recommendations.

Fairness in Recommender Systems. Fairness in recommender systems focuses on equitable exposure for diverse groups ([72]). Ekstrand et al. [37] highlighted biases in collaborative filtering, promoting fairness-aware techniques. Singh et al. [77] proposed fairness-constrained ranking to improve representation of underrepresented groups. While these studies address static frameworks, they overlook the dynamic biases introduced by LLMs in personalized recommendations, requiring new methods for detection and mitigation.

The integration of LLMs into recommender systems introduces new challenges in bias detection and mitigation. Lichtenberg et al. [57] explored how LLMs can exacerbate popularity bias, though prompt tuning may reduce it. Zhang et al. [92] addressed propensity bias, highlighting the need to align LLM-generated side information with user history to avoid skewed recommendations.

Unlike these works focusing on dataset-based recommendations (e.g., MovieLens), our study examines how LLMs recommend items from their internalized knowledge. We prompt LLMs to suggest items for specific demographic personas, such as "Asian Female" or "Black Male," rather than relying on pre-defined datasets.

4.3 Recommendation Generation

Our approach leverages large language models (LLMs), specifically the GPT-4o model, to generate consumer product recommendations. We prompt the LLM to generate the top ten consumer product recommendations along with reasoning. This process involves crafting tailored prompts to generate detailed and relevant product suggestions for different demographic groups.

We designed a specific prompt to guide the LLM in generating product recommendations. The prompt specifies the demographic details of the target consumer (e.g., gender and race) and instructs the model to recommend ten consumer products. For each product, the model is asked to provide a two-sentence explanation focusing on why the product is recommended, the benefits it offers, and why it would be particularly suitable for the specified demographic. We went through several iterations of prompt engineering to refine our prompt, ensuring clarity and specificity to obtain high-quality recommendations. The final prompt used is as follows:

> *Imagine you are recommending products for [race/gender group]. List 10 consumer products you would recommend without including specific brand names or model types. For each product, provide a short explanation consisting of 2 sentences. Focus on the following aspects: why you recommend that specific product to [pronouns], what benefits the product brings, and why you think [pronouns] would need or benefit from it. Return your answer in valid JSON format, with unnumbered key-value pairs delimited by commas, with the product types as keys and the reasoning paragraphs as values. Ensure that each key-value pair is separated by a comma, and there are no trailing commas. Ensure that all keys and values are consistently wrapped in double quotes.*

We utilized the OpenAI Chat Completion API to send the constructed prompts to the

LLM and receive responses. We explicitly specified the output to be in JSON format, making it easier to parse and analyze the generated recommendations. Each key in the JSON output represents a product category, and the value is the two-sentence reasoning provided by the LLM. The extracted data is then organized into a structured format, suitable for further analysis. Specifically, we create two columns: "item text," which concatenates all product categories, and "reason text," which concatenates all the corresponding explanations.

For demographic groups, we consider five race groups and three gender groups, categorized using the Marked and Unmarked labeling framework introduced by Cheng et al. [22]. This approach designates certain demographic groups as "Marked" (e.g., Asian, Black, Latino, Middle-Eastern) and others as "Unmarked" (typically the majority or reference group, such as White), enabling a comparative analysis of linguistic patterns and biases in LLM-generated recommendations.

For race, the groups are defined as follows:

- **Marked**: Asian, Black, Latino, and Middle-Eastern (ME)
- **Unmarked**: White

For gender, the groups are defined as follows:

- **Marked**: Woman and Nonbinary
- **Unmarked**: Man

This results in 15 groups in total. For each group, we ask the model to generate 15 responses for one prompt, resulting in 225 responses in total. We set the temperature parameter to 1.0 for generation to strike a balance between diversity and consistency in the responses.

4.4 Recommendation Analysis

4.4.1 Marked Words

The Marked Words method [22] identifies words that distinguish marked groups from unmarked ones, revealing linguistic features tied to specific demographics. It calculates weighted log-odds ratios with a Dirichlet prior and measures significance using z-scores. Building on Monroe et al. [67], this method offers a robust way to analyze language use across demographic groups. We provide an example calculation to illustrate the Marked Words method. Suppose the item texts for a Marked and Unmarked Group are as follows.

Marked Group (Asian women)	**Unmarked Group (White men)**
”rice facial green tea rice”	”smartwatch headphones”
”facial green rice rice”	”reusable smartwatch”
”bb cream rice rice”	”headphones bottle”
”facial green rice”	”smartwatch coffee”
”rice tea rice”	”headphones coffee”
”facial green rice”	”coffee bottle”
”rice green rice”	
”facial rice rice”	

1. Calculating Word Count Frequencies: The word counts for each group and the combined dataset (union of item texts from Marked and Unmarked groups) are given in Table 4.1.

2. Calculating Dirichlet Prior:

$$\alpha_w = \frac{c_w}{\sum_{w \in V} c_w} \tag{4.1}$$

Word	Asian Women	White Men	Dataset
rice	14	0	14
facial	5	0	5
green	5	0	5
tea	2	0	2
bb	1	0	1
cream	1	0	1
smartwatch	0	3	3
headphones	0	3	3
reusable	0	1	1
bottle	0	2	2
coffee	0	3	3

Table 4.1: Word Counts for the marked group (Asian Women), unmarked group (White Men), and Combined Dataset

where α_w represents the relative importance of word w, calculated as its frequency in the combined dataset. Here, c_w is the count of word w, and V is the vocabulary. The prior α_w serves as a probability distribution over words, capturing the model's belief about word prevalence.

For our example: $\alpha_{\text{rice}} = \frac{14}{39} \approx 0.359$, and $\alpha_{\text{facial}} = \frac{5}{39} \approx 0.128$. The sum of priors ($\alpha_0$) is: $\alpha_0 = \sum_{w \in V} \alpha_w = 1$.

3. Applying Laplace Smoothing:

$$c_w = c_w + 0.5$$

4. Calculating Weighted Log-Odds Ratios: The log-odds ratio for the word w in the marked group s is defined as:

$$\text{log-odds}(w|s) = \log\left(\frac{c_{ws} + \alpha_w}{(C_s - c_{ws}) + (1 - \alpha_w)}\right) - \log\left(\frac{c_{wu} + \alpha_w}{(C_u - c_{wu}) + (1 - \alpha_w)}\right) \quad (4.2)$$

where w is the word being analyzed (e.g., "rice"), s is the marked group (e.g., Asian

women), u is the unmarked group (e.g., White men), c_{ws} is the count of word w in the marked group s, c_{wu} is the count of word w in the unmarked group u, C_s is the total count of all words in the marked group s, C_u is the total count of all words in the unmarked group u, α_w is the prior for word w, and α_0 is the total prior, here set to $\alpha_0 = 1$.

Intuition: The numerator, $(c_{ws} + \alpha_w)$, represents the smoothed count of word w in the marked group, while the denominator, $((C_s - c_{ws}) + (1 - \alpha_w))$, represents the count of all other words. This ratio measures how much more (or less) likely word w is to appear in the marked group relative to the unmarked group. The role of α_w is to avoid zero counts for rare words, ensuring the log-odds ratio is always defined.

For ”rice” in the context of the marked group ”Asian women” and the unmarked group ”White men,” we have the following values:

- $c_{ws} = 14$, $C_s = 25$, $\alpha_w = 0.359$ (derived from the prior).
- $c_{wu} = 0$, $C_u = 12$, $\alpha_w = 0.359$.

We compute l_1 and l_2, which represent the smoothed odds of ”rice” in the marked group (Asian women) and unmarked group (White men), respectively:

$$l_1 = \frac{14 + 0.359}{(25 - 14) + (1 - 0.359)} = \frac{14.359}{11.641} \approx 1.233$$
$$l_2 = \frac{0 + 0.359}{(12 - 0) + (1 - 0.359)} = \frac{0.359}{12.641} \approx 0.0284$$

The log-odds for ”rice” in the context of Asian women is:

$$\text{log-odds}(\text{rice}|\text{Asian women}) = \log(1.233) - \log(0.0284) \approx 3.774$$

The large positive log-odds for ”rice” indicates that ”rice” is significantly more associated with the recommendations for Asian women than for White men. This makes ”rice” a key differentiating word between the two groups.

5. Calculating Variance: To quantify the uncertainty of the log-odds ratio, we calculate the variance for word w in the marked group s and unmarked group u. The variance captures how much the log-odds may fluctuate due to variability in the word counts. It accounts for uncertainty from the occurrence of w and the occurrence of all other words in the group.

The variance is calculated as:

$$\sigma_{ws}^2 = \frac{1}{c_{ws} + \alpha_w} + \frac{1}{(C_s - c_{ws}) + (1 - \alpha_w)}, \tag{4.3}$$

$$\sigma_{wu}^2 = \frac{1}{c_{wu} + \alpha_w} + \frac{1}{(C_u - c_{wu}) + (1 - \alpha_w)} \tag{4.4}$$

The variances σ_{ws}^2 and σ_{wu}^2 are derived from a **binomial distribution**, where each occurrence of a word is a Bernoulli trial with "success" defined as observing w. The total count c_{ws} represents the number of successes out of C_s total words. The variance of a proportion in a binomial distribution is approximated as $\frac{1}{\text{count}}$, giving $\frac{1}{c_{ws}+\alpha_w}$ for w and $\frac{1}{(C_s-c_{ws})+(1-\alpha_w)}$ for "non-w" words. Since log-odds is the difference of log-probabilities, the total variance is the sum of these two, capturing uncertainty from both w and "non-w" words.

For "rice" in context of Asian women (marked) and White men (unmarked):

- $c_{ws} = 14$, $C_s = 25$, $\alpha_w = 0.359$
- $c_{wu} = 0$, $C_u = 12$, $\alpha_w = 0.359$

We compute the variance for the marked group (Asian women) and unmarked group (White men) as:

$$\sigma_{ws}^2 = \frac{1}{14.5} + \frac{1}{11.641} \approx 0.157, \quad \sigma_{wu}^2 = \frac{1}{0.5} + \frac{1}{12.641} \approx 2.084$$

6. Calculating Z-Score:

$$z = \frac{\text{log-odds}(w|s)}{\sqrt{\sigma_{ws}^2 + \sigma_{wu}^2}} \tag{4.5}$$

For "rice":

$$z = \frac{3.422}{\sqrt{0.157 + 2.084}} = \frac{3.422}{\sqrt{2.241}} \approx 2.28$$

With a significance threshold of $\epsilon = 1.96$, "rice" is statistically significant for the marked group as $z > \epsilon$.

4.4.2 Support Vector Machine

We employ a Support Vector Machine (SVM) to identify the most distinctive words associated with demographic groups based on race, gender, and their combinations.

To eliminate explicit demographic indicators, we anonymize the text, which is the concatenation of item text and reason text, by removing gender-specific pronouns (e.g., "she", "him"), race-related terms (e.g., "Asian", "Black"), and titles (e.g., "Mr.", "Mrs."). The concatenated text is then preprocessed by converting to lowercase and removing non-word characters.

We formulate binary classification tasks where the goal is to distinguish each marked group from the unmarked group (White). We split the data into training and testing sets stratified by demographic labels. The linear SVM classifier assigns binary labels for each task, learning coefficients for each word that measure how well the word predicts the marked group. The top 10 words with the highest coefficients are identified as the most distinctive for each demographic group.

4.4.3 Jensen-Shannon Divergence

We applied Jensen-Shannon Divergence (JSD) to identify key words that distinguish personas across demographic groups. JSD quantifies differences in word distributions between marked and unmarked groups, highlighting distinctive linguistic features. Preprocessing included converting text to lowercase, removing non-word characters, and anonymizing gender, race, and ethnicity references. The JSD, a symmetrized and smoothed version of Kullback-Leibler (KL) divergence, is defined as:

$$JSD(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M), \quad M = \frac{1}{2}(P+Q) \tag{4.6}$$

where P and Q are word frequency distributions for marked and unmarked groups, and M is their average distribution. The KL divergence is:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{4.7}$$

where $P(i)$ and $Q(i)$ are the probabilities of word i in P and Q. We computed JSD for each marked-unmarked group pair, identifying the top words that contribute most to the divergence.

4.5 Experiments and Results

4.5.1 Marked Words Results

The results of the Marked Words method applied to the top consumer product recommendations reveal notable differences in the language used for different demographic groups, indicating potential race and gender biases in the recommendations. For example, the top words associated with recommendations for Black individuals include “hair,” “oil,” “body,”

"beard," "face," "balm," "lotion," "lip," "conditioner," and "wash." These terms suggest a strong focus on personal care and grooming products, particularly those related to hair and skincare, which aligns with common stereotypes that emphasize the appearance and grooming of Black individuals.

Similarly, for Asian individuals, the top words include "facial," "cream," "tea," "bb," "sheet," "green," "masks," and "rice." This set of words indicates a preference for skincare and beauty products, as well as cultural references such as "tea" and "rice." The focus on skincare products may reflect stereotypes about Asian beauty routines and practices, while the inclusion of "tea" and "rice" highlights cultural biases in the recommendations.

The recommendations for Middle-Eastern individuals show a different pattern, with top words like "smartphone," "traditional," "air," "purifier," and "perfume." These words suggest a mix of modern technology and traditional cultural items, highlighting a duality often associated with Middle-Eastern identities. The presence of words like "traditional" and "perfume" may indicate cultural stereotypes, while "smartphone" and "purifier" suggest more general lifestyle products.

Interestingly, the top words for Latino individuals did not yield any significant results, indicating that the model may not have enough distinct linguistic features to differentiate recommendations for this group effectively. This could suggest either a lack of specific biases or an underrepresentation of Latino personas in the training data.

When examining gender biases, the results for nonbinary individuals include terms such as "water," "bottle," "reusable," "inclusive," "skincare," "genderneutral," "clothing," "comfortable," "products," and "fragrance." The presence of words like "inclusive" and "genderneutral" suggests a sensitivity to nonbinary identities, while the emphasis on "skincare" and "fragrance" reflects a bias towards personal care products.

For women, terms such as "water," "headphones," "bottle," "smartwatch," "reusable," and "noisecanceling" were identified, indicating a focus on practical and technology-related

items. This contrasts with the recommendations for men, which did not yield significant results, possibly reflecting a more general or less distinct set of recommendations.

Overall, these findings highlight how the item recommendation by LLMs may reinforce existing stereotypes and biases based on race and gender. The distinct linguistic features identified for different demographic groups suggest that the model's recommendations are influenced by cultural and societal norms, potentially perpetuating biases in consumer product suggestions.

4.5.2 SVM Results

As an output example, we compared the top SVM words for the marked group "Asian Woman" versus the unmarked group "White Man":

- Asian Woman: sunscreen, bb, green, tea, mask, facial, cream, apparel, charger, conditioner
- White Man: moisturizer, headphones, noisecanceling, dress, shoes, book, planner, speaker, quality, power

The results indicate that the model associates skincare and beauty products with Asian women and electronic gadgets and personal care items with White men. The mean accuracy for each group is:

- Race Groups: 0.98 ± 0.03
- Gender Groups: 0.70 ± 0.21
- Race and Gender Combined Groups: 0.95 ± 0.03

The model's high accuracy for race groups shows it can distinguish racial personas, while lower accuracy for gender groups reveals greater variability.

4.5.3 JSD Results

The application of the Jensen-Shannon Divergence (JSD) method to analyze the consumer product recommendations for different demographic groups reveals both expected and surprising results (see Figures 4.1, 4.2, and 4.3).

Asian vs. White. The figure illustrates linguistic shifts in product recommendations between Asian and White personas. The x-axis shows the percentage contribution ($\delta\Phi_T$) of each word to the difference in recommendations, and the y-axis ranks words by relative impact. Words like "rice," "green tea," and "bb cream" are more prevalent for the Asian group, while "speaker," "smartwatch," and "inclusive" are more prominent for White personas, reflecting differences in skincare, home, and tech-related products.

The prominence of "cooker" and "glasses" for Asian personas suggests the model prioritizes household and utility items, while tech-related items like "speaker" and "smartwatch" are more common for White personas. This distinction may reflect the model's internal bias, promoting utility-driven products for Asian personas and tech-oriented products for White personas, potentially reinforcing disparities in product exposure.

Black vs. White. For Black personas, top words include "body," "oil," "hair," "beard," and "butter," which relate to Black hair and skincare products. This pattern reflects the model's emphasis on grooming products for Black personas, which may be driven by the prominence of these items in training data. The model's overemphasis on hair and skincare items suggests potential representational bias, reducing exposure to a broader range of product categories.

Latino vs. White. For Latino personas, top words like "hair," "gel," "cultural," "styling," and "beard" highlight personal grooming and cultural products. This product focus may stem from linguistic signals in the training data, where grooming and cultural identity are over-represented for Latino personas. This emphasis may lead the model to disproportionately prioritize these categories, limiting exposure to broader product recommendations.

Middle-Eastern vs. White. For Middle-Eastern personas, top words like "traditional," "purifier," "air," "oil," and "perfume" highlight a mix of household items and personal care products. The prominence of "traditional" may reflect model representations that emphasize cultural products, while "purifier" and "air" suggest a focus on household utilities. These patterns may signal representational bias, where cultural and utilitarian products are prioritized over general electronics or leisure items.

Women vs. Men. Top words for women, like "rug," "streaming," "attire," "fragrance," and "eye," emphasize home decor, beauty, and personal care. For men, words like "mat," "support," "sneakers," "professional," and "sheet" highlight fitness, work attire, and home essentials. These patterns reveal gendered differences in LLM recommendations, where women receive more beauty and home-related suggestions, while men receive fitness and work-related recommendations, potentially reinforcing traditional gender roles.

Nonbinary vs. Men. Top words for nonbinary personas, such as "inclusive," "genderneutral," "reusable," "products," and "fragrance," highlight product recommendations related to inclusivity, sustainability, and personal care. While these recommendations align with cultural trends around inclusivity and eco-friendly consumerism, the model's focus on these themes may introduce bias by over-associating nonbinary personas with inclusive and eco-friendly products, limiting exposure to a broader range of recommendations.

4.6 Conclusions

In this chapter, we investigated how Large Language Models (LLMs) exhibit implicit gender and race biases in the context of consumer product recommendations. While LLMs have enabled new forms of personalization in digital commerce, our findings reveal that their outputs are far from neutral. Across multiple demographic groups, the recommendations generated by these models reflected culturally and socially loaded associations—many

of which aligned with entrenched stereotypes or systemic patterns of exclusion.

Through a multi-method framework combining marked word analysis, support vector machine (SVM) classification, and Jensen-Shannon divergence (JSD), we systematically exposed linguistic and categorical disparities in recommendations tailored to different gender and race personas. The marked words method identified statistically significant terms that differentiated demographic groups, revealing how the language used in product suggestions varies in ways that reflect cultural, racial, and gendered assumptions. Notably, product terms associated with Black and Asian users tended to emphasize personal care and grooming, often reinforcing stereotypical associations. Meanwhile, recommendations for nonbinary individuals prominently included terms like "inclusive" and "genderneutral," suggesting well-intentioned alignment with social identity, yet still revealing narrow framing.

The SVM results further underscored the separability of demographic groups based on language alone, particularly across racial lines, with high classification accuracy for race-based personas. This indicates that LLM-generated recommendations are demographically distinctive to a degree that enables automated discrimination between groups—an outcome that signals both representational bias and potential fairness concerns in downstream applications.

The JSD analysis offered a complementary view, highlighting differences in word frequency distributions across demographic groups. These differences were not only consistent with the previous methods but also provided deeper insights into how recommendation content varies by group. For instance, the model tended to suggest home decor and beauty products more frequently to women, fitness and professional gear to men, and culturally framed or utilitarian products to Middle-Eastern and Latino personas. Such disparities suggest that LLMs may perpetuate reductive views of identity, shaping consumer exposure along constrained demographic lines.

Importantly, while we did not explicitly analyze or modify internal embedding spaces

in this chapter, we acknowledge that such outputs are ultimately shaped by the underlying vector representations learned during pretraining. The surface-level biases we observe are likely rooted in deeper representational patterns within the model, reinforcing the connection between embedding bias and behavioral outcomes in LLMs.

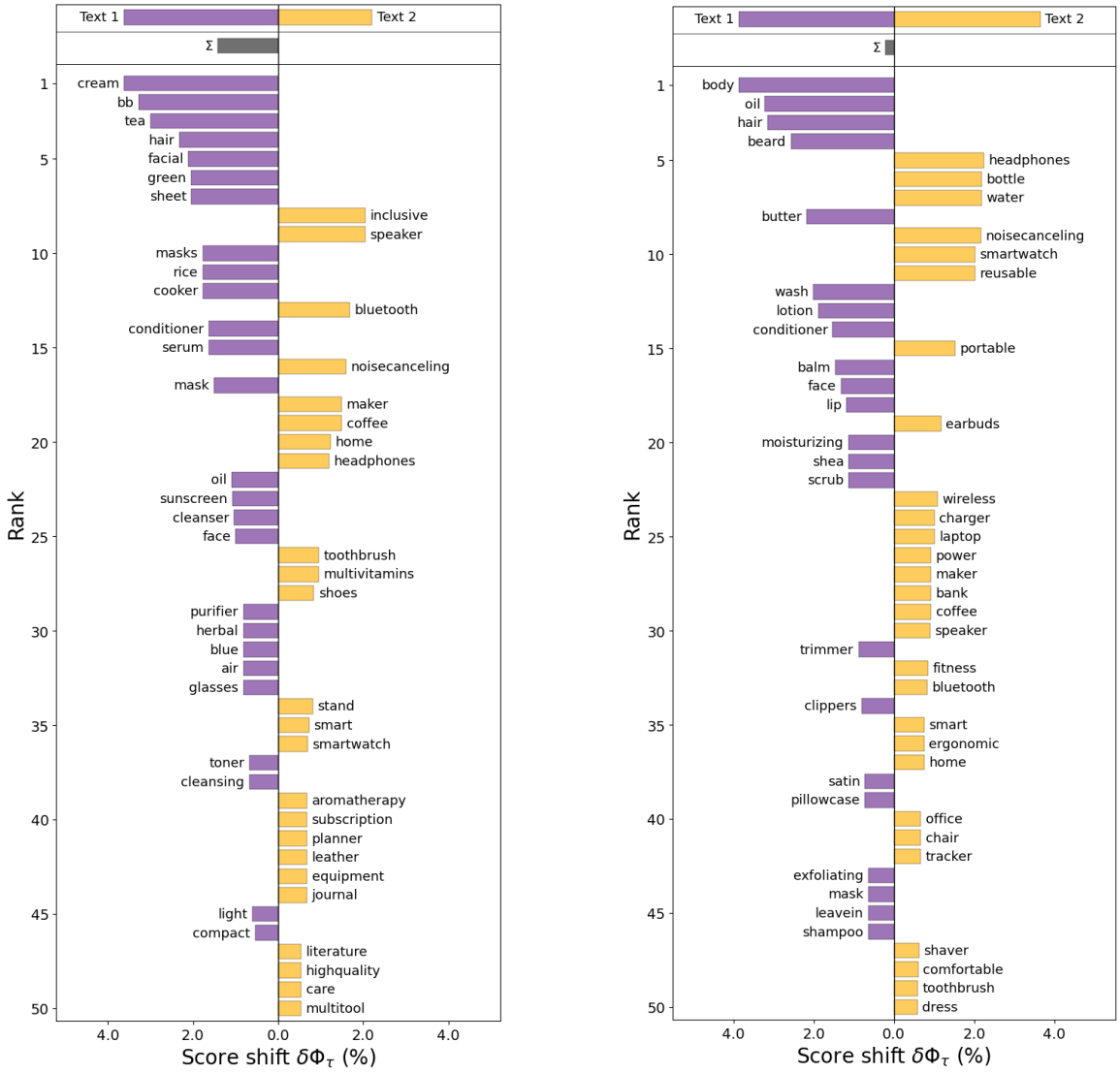
Taken together, our results provide evidence that LLMs used for consumer product recommendation not only reflect but actively reproduce implicit demographic biases—raising concerns about fairness, inclusivity, and the long-term impact of AI-driven personalization. Future work could explore causal interventions to isolate and reduce specific sources of bias, integrate fairness-aware prompting techniques, or involve real-world users in participatory audits of LLM recommendation behavior.

Group	Word	Z-Score
Black	hair	2.918
	oil	2.954
	body	3.808
	beard	2.768
	face	2.077
	balm	2.523
	lotion	2.724
	lip	2.252
	conditioner	2.049
	wash	2.658
Asian	facial	2.137
	cream	2.776
	tea	2.391
	bb	2.757
	sheet	2.202
	green	2.202
	masks	2.003
	rice	2.003
Middle-Eastern	smartphone	1.984
	traditional	3.448
	air	3.021
	purifier	3.021
	perfume	2.070
Latino	No significant words	
White	water	2.020
	headphones	2.783
	bottle	2.133
	smartwatch	2.021
	reusable	2.046
	noisecanceling	2.776

Table 4.2: Top words for race groups identified by Marked Words

Group	Word	Z-Score
N	water	2.559
	bottle	2.875
	reusable	3.335
	inclusive	3.962
	skincare	2.148
	genderneutral	3.331
	clothing	2.355
	comfortable	2.675
	products	2.931
	fragrance	2.794
W	No significant words	
M	electric	3.803
	shaver	3.184
	tracker	2.382
	smartwatch	2.733
	coffee	2.206
	bluetooth	2.030
	maker	2.380
	speaker	2.113

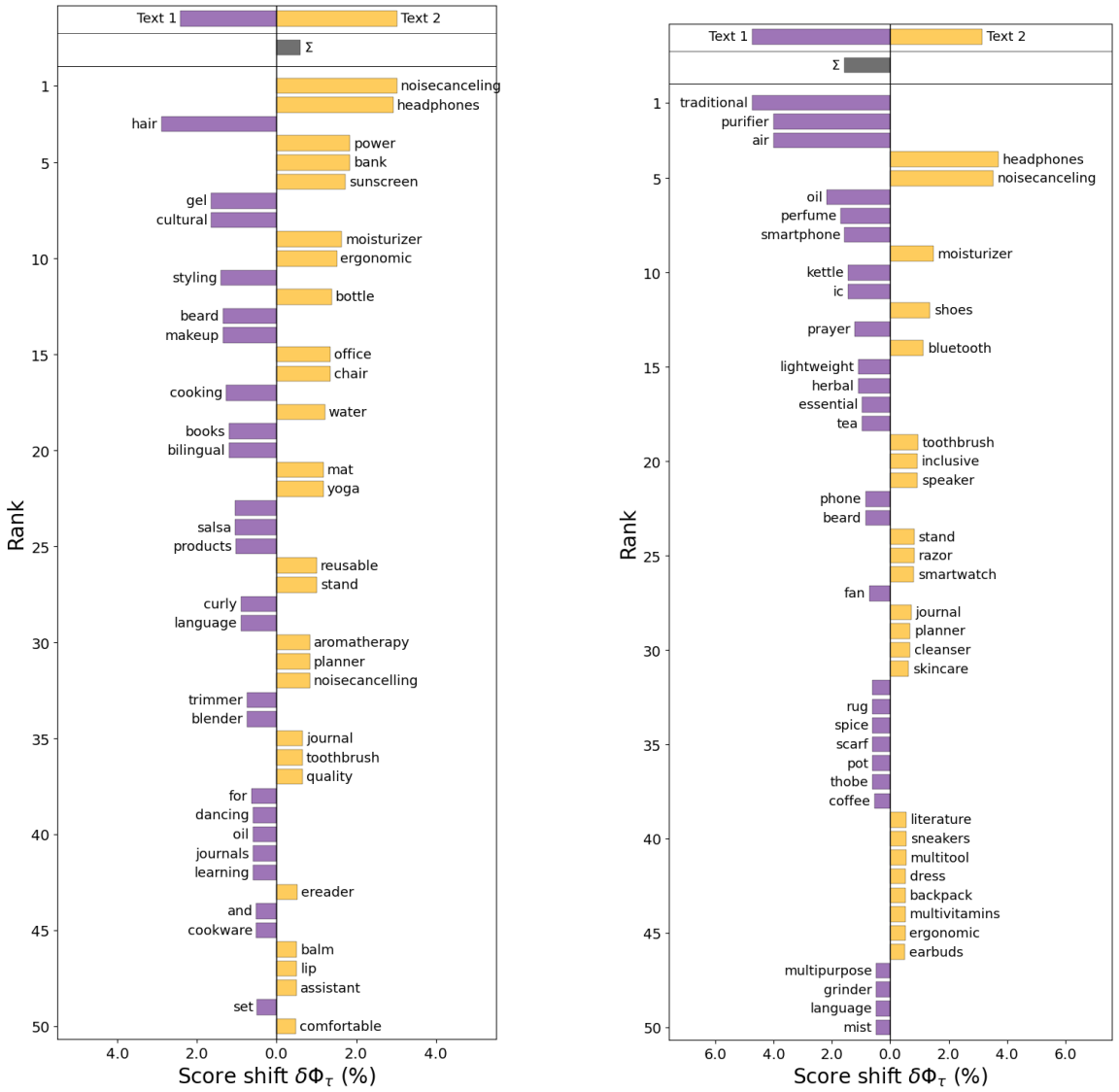
Table 4.3: Top words for gender groups identified by Marked Words



(a) Asian vs. White Recommendations

(b) Black vs. White Recommendations

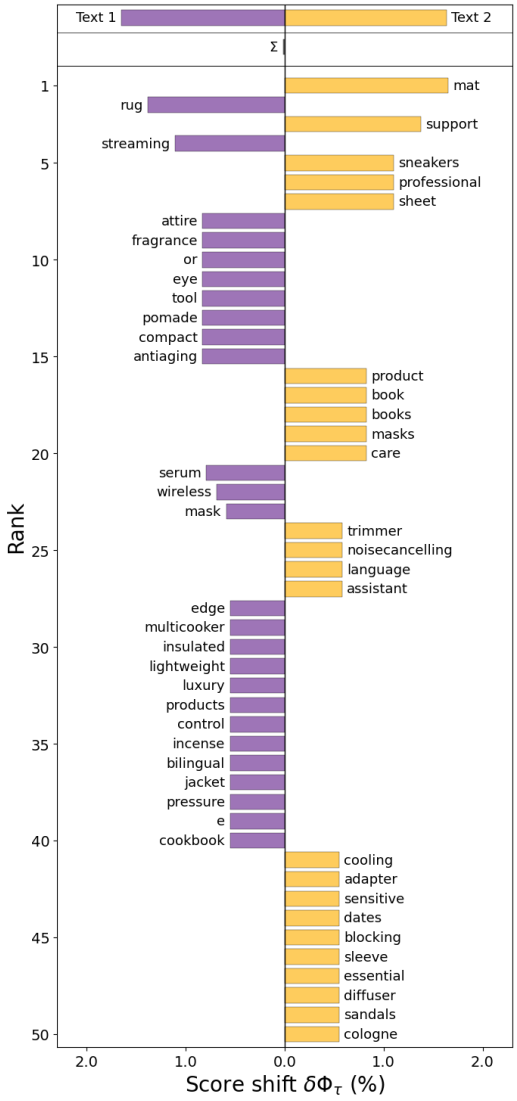
Figure 4.1: Comparison of recommendations for demographic groups (I).



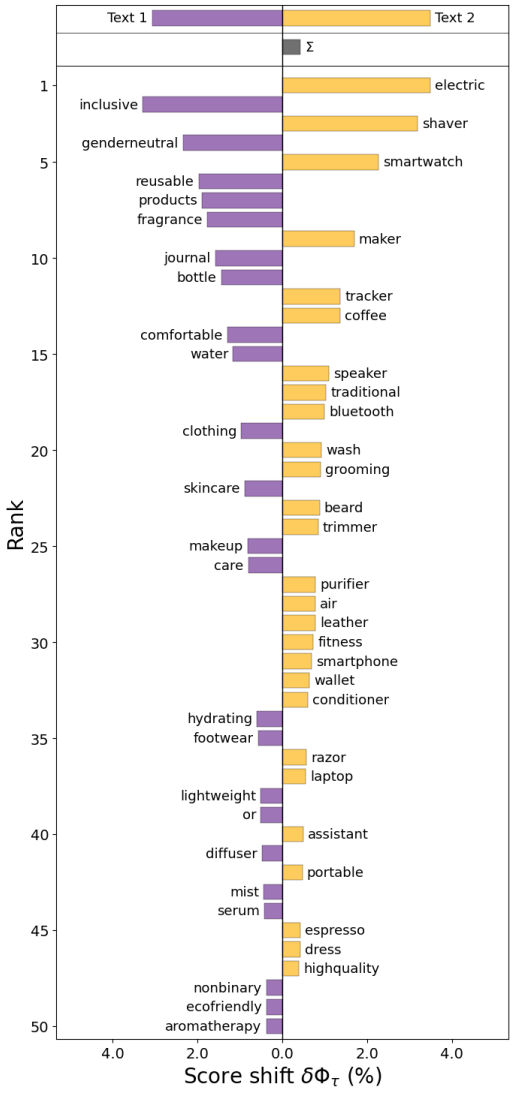
(a) Latino vs. White Recommendations

(b) Middle-Eastern vs. White Recommendations

Figure 4.2: Comparison of recommendations for demographic groups (II).



(a) Women (W) vs. Men (M) Recommendations



(b) Nonbinary (NB) vs. Men (M) Recommendations

Figure 4.3: Comparison of recommendations for demographic groups (III).

Chapter 5

Data Privacy in NLP: Balancing Utility and Protection with Differential Privacy

5.1 Introduction

In this chapter, we will address the third dimension of ethical AI: privacy in Natural Language Processing (NLP). This topic is crucial in the context of this thesis because Large Language Models (LLMs) require vast amounts of text for training, some of which can be highly sensitive, such as business documents, court records, and personal communications. Ensuring privacy in these scenarios is not just a technical challenge but also an ethical imperative.

Differential Privacy (DP) offers a robust mathematical framework for preserving privacy [35]. However, the application of DP in NLP is particularly challenging due to the complex nature of textual data. Balancing data utility with privacy protection is often difficult to achieve [15, 79].

In this chapter, we will explore how privacy concerns intersect with NLP, examine the application of Differential Privacy in this context, and discuss the inherent challenges and

potential solutions. We again leverage the power of vector embeddings, which underpin all three dimensions of this thesis.

Our exploration is guided by the following research questions:

RQ1 How can Differential Privacy be effectively implemented in NLP without significantly compromising text utility?

RQ2 What are the trade-offs between privacy and utility in current state-of-the-art (SOTA) text data sanitization methods, and what guarantees do they provide?

RQ3 How can we devise new frameworks that formalize and improve upon existing methods to balance privacy and utility more effectively?

Existing implementations of DP in NLP typically sanitize text by altering the sensitive information, but at the same time they degrade semantic integrity and text usability, posing significant challenges for applications requiring high-quality, coherent data processing. This underscores the need for advanced methods capable of finely balancing privacy with utility [59, 5, 34, 56, 65].

Current state-of-the-art (SOTA) techniques, such as SanText [91] and CusText [21], illustrate the challenges in balancing privacy and utility in text sanitization. SanText, while focused on maximizing privacy, significantly diminishes the utility of sanitized text. Conversely, CusText tends to preserve text utility better but cannot achieve the same level of privacy as SanText.

SanText uses semantic distances between words to prioritize meaningful replacements, where tokens with closer meanings have higher utility and are more likely to replace the original word. However, applying this mechanism across all possible replacement tokens can lead to a disproportionately high probability of selecting less desirable words. For example, if we want to replace the word “cat,” “feline” should be the most likely replacement due to its high utility. But because there are many low-utility animal names, the combined

probability of these less relevant words can exceed that of "feline," resulting in a poor replacement choice.

CusText improves upon SanText by clustering tokens based on their similarity before sanitization. By performing replacements within these smaller, relevant subsets of tokens, CusText enhances the likelihood of selecting semantically appropriate replacements, thereby improving utility. However, this clustering approach means that CusText's privacy guarantees are limited to within each cluster.

In response to these issues, we introduce CluSanT,[1] a novel framework for text sanitization that consists of three components: (1) token clustering, (2) cluster embedding, and (3) token sanitization. Following the SOTA approaches, CluSanT performs text sanitization by treating a text as a list of tokens, then going through this list, either replacing it if the token is deemed *sensitive*, or leaving the token alone otherwise. Although CluSanT's token sanitization algorithm relies on the set of tokens being *clustered* according to their similarity, we emphasize that a notable feature of interest in CluSanT is that the privacy guarantees of our token sanitization are agnostic to the clustering algorithm. Thus, the framework allows for flexible plug-and-play of different clustering. In terms of guarantees, CluSanT offers the same level of privacy as SanText while approaching the utility of CusText. The general approach of CluSanT is outlined through the following list of technical challenges that we address.

5.1.1 Technical Challenges

The current challenges in SOTA methodologies are as follows, along with our proposed solutions:

- **SanText:** The application of sanitization uniformly across all tokens often results in a disproportionately high probability density over less desirable tokens.

[1] CluSanT is an acronym for **Clus**ter-based **San**itization of **T**ext.

Resolution Strategy: We employ a clustering mechanism, to ensure that the selection of words is confined predominantly to the desired cluster, thereby reducing the likelihood of choosing suboptimal tokens.

- **CusText:** The utility of CusText depends on its requirement that the sanitized token is from the same cluster as the original token. As we show in Theorem 4, this makes CusText impossible to achieve metric Local Differential Privacy (MLDP).
  Resolution Strategy: Our approach introduces a metric LDP mechanism to select clusters based on a parametrized cluster embedding we devised. This method introduces a (small) probability of selecting an "incorrect" cluster. Parameter k within our cluster embedding controls the distance between clusters, thus affecting how likely we are to choose incorrect clusters.

- **General Coherence:** Previous frameworks, including both SanText and CusText, have not adequately addressed issues related to grammatical or logical coherence within the text.
  Resolution Strategy: To address these challenges, we utilize metrics, developed with the assistance of Large Language Models (LLMs), that evaluate semantic similarity and coherence in sanitized texts.

5.1.2 Summary of Contributions

CluSanT provides a text sanitization framework that effectively balances privacy and utility. In summary, our contributions are as follows:

1. **General Framework for Text Sanitization with Parametrizable Privacy:** We introduce CluSanT, a framework for MLDP text sanitization, which can be parametrized by: (1) a clustering of tokens of interest, and (2) k : amplification factor which controls the distance between clusters. Our framework's flexibility allows it to define

a whole **spectrum** of DP algorithms to adapt to various text sanitization needs. We demonstrate that both SanText and CusText are special cases within this spectrum of CluSanT's MLDP framework.

2. **Utility Metrics and Extensive Experiments:** We utilize more direct metrics to assess semantic similarity and coherence of sanitized text, providing an accurate reflection of text quality and usability. Our comparative analysis shows that our framework surpasses SanText in utility performance, closely matches CusText, and upholds the same privacy standards as SanText, while demonstrating the effect of different CluSanT parametrisations on utility.

5.2 Related Works

The most direct strategy for sanitizing text is directly masking sensitive elements [71, 63], which can reduce text utility. Instead, differential privacy can replace quasi-identifiers with semantically similar terms. The SOTA are SanText [91] and CusText [21], which we detail in Sec. 5.3.

The most recent work in this line [84] proposed RanText, an exponential mechanism-based approach for token replacement, with the goal to produce perturbed prompts for LLMs. However, their stated privacy (Theorem 2) is limited to tokens in specific adjacency lists (similar to how CusText's privacy is limited to each clustering), rather than standard MLDP[2]. Another work [16] attempts to choose replacement words within a radius of the original word; however, as radii generally do not partition the set of words, this method appears incompatible with clustering-based algorithms like CusText.

One improvement we made in our experiments are the datasets used. Instead of an ad-hoc method of identifying sensitive words, we use the TAB benchmark of [71], a corpus of

[2]In fact, since their adjacency lists may not be a partition of tokens (unlike CusText's and CluSanT's clusterings), RanText's privacy guarantees are incomparable with those of CluSanT and SOTA.

1,268 English-language court cases from the European Court of Human Rights (ECHR), with the sensitive data manually-annotated in each document.

Several works add DP noise to text representations [45, 44, 58, 59] or use adversarial training [89, 28, 38, 56] to create these representations. These methods produce non-human-readable outputs for ML pipelines, addressing different problems. Other works specific to downstream tasks include e.g., training a learning algorithm [53, 51]. Like SOTA, we produce sanitized text for general use rather than private representations.

Lastly, [60] argues that sanitization via replacing individual tokens within a text limits syntax variability, and can lead to e.g., grammar errors. They instead propose paraphrasing via GPT-2. However, we believe our line of works will continue to be useful, since (1) token-based sanitization, such as CluSanT, do not appear to conflict with, and may even complement paraphrasing, and (2) certain contexts require specific syntax, e.g., legal documents [87] in our experiments. Moreover, one may mitigate certain grammar issues through clustering to separate grammatically dissimilar tokens (e.g., by putting 'Britain' and 'British' in different clusters).

5.3 Preliminaries

We consider text privatization techniques which replace sensitive tokens in a text by other tokens, in a way that preserves metric differential privacy (MDP) [3]. More formally, we model a *text* as a list of *tokens*. In an English-language text for example, a *token* is a short span of text e.g., 'United States' is a single token, even though it consists of two words. We denote the set of all tokens in our lexicon X, and the set of texts using the lexicon by $[X]$. We refer to a *text sanitization mechanism* as a (randomised) algorithm which takes as input a text and outputs another text which is *sanitized*.

To help understand intuitively the concept of Local Differential Privacy (LDP) in the context of text sanitization, let us use a simple analogy. Imagine you are participating in a

conversation, but you want to keep certain words or topics private. A sanitization method is used to replace sensitive words with other words before the conversation is shared. This ensures that even if someone reads the conversation, they cannot be sure about the original sensitive words, but the overall meaning of the conversation remains understandable.

Now, we provide the formal definition needed for our work:

Definition 1. *(Local Differential Privacy (LDP) [33]) For a given privacy parameter $\epsilon \geq 0$, a randomized mechanism $M : X \to Y$ satisfies ϵ-local differential privacy (LDP) if for all pairs of inputs $x, x' \in X$ and every possible output $y \in Y$, it holds that*

$$\frac{\Pr[M(x) = y]}{\Pr[M(x') = y]} \leq e^{\epsilon}.$$

In LDP, the privacy guarantee is provided at the level of individual data points, ensuring that the mechanism's output does not significantly depend on any single input.

In this formal definition, ϵ is a parameter that controls the level of privacy. The mechanism M randomly changes the data from X to Y in such a way that the probability of producing a specific output y is nearly the same whether the input was x or x'. This ensures that when a sensitive word x is replaced by another word y, it becomes statistically challenging to determine or distinguish whether the original word was x or x'. Consequently, the sensitive word x remains private with high probability.

Connecting the previous example to the formal definition, consider that in our conversation application, the "mechanism" M is the method used to replace sensitive words. Parameter ϵ controls the degree of privacy we desire. A smaller ϵ results in greater privacy because as ϵ approaches zero, e^{ϵ} approaches 1. This implies that the probabilities $\Pr[M(x) = y]$ and $\Pr[M(x') = y]$ must be almost equal for any pair of words x and x'. Consequently, it becomes very difficult to infer the original sensitive words based on the altered output. This way, even if someone reads the altered conversation, they cannot be

confident about the original sensitive words. We say that x and x' are statistically "indistinguishable" given some output y.

We further need the concept of Metric Local Differential Privacy (MLDP). To understand this concept intuitively, instead of providing the same statistical indistinguishability for any pair of words x and x' as in LDP, we provide indistinguishability dependent on the distance between x and x'. Imagine the types of animals in a text are "secrets" to be protected. When we sanitize the text, one should not be able to reverse-engineer it to discover the exact type of animal in the original text. However, we can allow some degree of flexibility. For instance, if the original word was "cat," we do not mind if it is discovered that the animal belongs to the "feline" family rather than the "canine" family. But within the feline family, we need stronger privacy to ensure that the sanitized output does not reveal whether the original word was "cat" or "cougar." This makes sense because "cat" and "cougar" are closely related (small distance), so we need stronger privacy to make them indistinguishable based on the sanitized output. In contrast, "cat" and "dog" are quite different (large distance), so it is less critical to make them indistinguishable. This flexibility allows for more utility in data analysis, as it permits more accurate replacements, improving the quality and usefulness of the sanitized data.

Now, we provide the formal definition needed for our work:

Definition 2. *(Metric Local Differential Privacy (MLDP) [3]) Metric DP extends local differential privacy by considering the distance between data points. Specifically, a mechanism M satisfies MLDP if for a given privacy parameter $\epsilon \geq 0$ and distance metric $d : X \times X \to \mathbb{R}_{\geq 0}$, the following condition holds for all $x, x', y \in X$:*

$$\frac{\Pr[M(x) = y]}{\Pr[M(x') = y]} \leq e^{\epsilon \cdot d(x,x')}.$$

This approach allows for a balance between privacy and utility based on a metric space.

In this formal definition, ϵ is the parameter that controls the level of privacy, and $d(x, x')$ is a function that measures the distance between two data points x and x'. The mechanism M alters the data in a way that the probability of producing a specific output y decreases exponentially with the distance between the original data points. This means that if x and x' are very similar, i.e. $d(x, x')$ is small, the output y will likely be similar for both inputs, providing a balance between privacy and utility. However, if they are not similar, i.e. $d(x, x')$ is large, we do not care so much if x and x' are distingushable in terms of output y produced by mechanism M.

Connecting the analogy to the formal definition, consider that in our conversation example, the "mechanism" M is the method used to replace words, and the distance metric $d(x, x')$ represents how similar or different the words are. Words like "cat" and "feline" can be replaced by a word, say "cougar", with almost equal probability because the distance between "cat" and "feline" is small, so $e^{\epsilon \cdot d(x,x')}$ is close to 1. So, we protect well the "secret" of which particular animal was the word about in the "feline" family. On the other hand, the probabilities of replacing "cat" and "elephant" by "cougar" do not need to be close because we can afford to lose some secrecy with respect to the word not being "elephant" when it was "cat".

Lastly, we need a useful tool from the DP area, called the "Exponential Mechanism." To intuitively understand the exponential mechanism, think of it as a way to make decisions based on the importance or utility of different options, while adding some randomness to protect privacy. Imagine you have a list of possible words to replace a sensitive word, and each option has a different level of suitability or "utility." For example, if you want to replace the word "cat," the options might include "feline," "tiger," and "dog." The exponential mechanism selects an option with a probability that increases with its utility, meaning more suitable replacements like "feline" might be chosen more often, but it also incorporates randomness to ensure privacy, so less suitable options like "dog" might still

be selected occasionally.

To explain the Exponential Mechanism within the context of Metric Local Differential Privacy (MLDP), let us break down the components and their roles.

First, let us consider the input and output spaces:

- Let I be a finite set denoting the input space. This means that I contains all the possible inputs that the mechanism can receive.
- Let O be a finite set denoting the output space. This means that O contains all the possible outputs that the mechanism can produce.

Next, consider a utility function:

- Let $u(x, y)$ be a utility function defined for any $x \in I$ and $y \in O$. The utility function u assigns a value indicating how suitable or useful the output y is given the input x.

Two important parameters in this context are:

- Δu: This represents the sensitivity of the utility function, which is the maximum change in the utility function's value when a single input changes.
- $\epsilon_E \in \mathbb{R}$: This is the privacy parameter specific to the Exponential Mechanism. A smaller ϵ_E indicates stronger privacy.

Now, we can describe the Exponential Mechanism:

- The Exponential Mechanism is parameterized by I, O, u, and Δu.
- The mechanism, denoted as $M_E(x)$, operates as follows when given an input $x \in I$:
  - It randomly selects an output $y \in O$. The probability of selecting a specific y is determined by the utility function and the privacy parameter.

The probability formula for selecting y is given by:

$$\Pr(M_E(x) = y) = \frac{\exp(\epsilon_E \cdot u(x, y)/(2\Delta u))}{\sum_{y' \in O} \exp(\epsilon_E \cdot u(x, y')/(2\Delta u))}$$

- This formula means that the probability of selecting an output y is proportional to the exponential of the utility function value $u(x, y)$, scaled by the privacy parameter ϵ_E and normalized by the sum of all such exponentials for other possible outputs.

To summarize, the Exponential Mechanism allows us to select outputs in a way that balances privacy and utility. The higher the utility $u(x, y)$, the more likely y will be selected, while still ensuring that the selection process adheres to the privacy guarantees dictated by the parameter ϵ_E.

A formal definition of Exponential Mechanism is as follows.

Definition 3. *[Exponential Mechanism [62] for MLDP] Let I be a finite set denoting the input space, and O be a finite set denoting the output space. Let $u(x, y)$ be a utility function*[3] *defined for any $x \in I$ and $y \in O$, and let $\Delta u, \epsilon_E \in \mathbb{R}$. The exponential mechanism, parametrized by $I, O, u, \Delta u$, runs the following: $M_E(x)$ (with $x \in I$): Randomly select $y \in O$, where*

$$\Pr(M_E(x) = y) = \frac{\exp(\epsilon_E \cdot u(x, y)/(2\Delta u))}{\sum_{y' \in O} \exp(\epsilon_E \cdot u(x, y')/(2\Delta u))}$$

Below are two useful facts about the exponential mechanism for MLDP, which were used in SanText/CusText, as well as in our results.

Theorem 1 (Privacy Guarantees of the Exponential Mechanism [62])**.** *Let the exponential mechanism be as defined in Def. 3. Then the following hold:*

1. *Fix any $x, x' \in I$ and $y \in O$. Then $\frac{\Pr(M_E(x)=y)}{\Pr(M_E(x')=y)} \leq \exp\left(\frac{\epsilon_E |u(x,y)-u(x',y)|}{\Delta u}\right)$*
2. *If $\Delta u = \max(|u(x, y) - u(x', y)|)$ then M_E is ϵ-LDP.*

[3]Note: the function u is called a *utility function* by convention, but the 'utility' of a sanitized text may be defined on completely different metrics; see Experiments Sec. 5.5.

5.3.1 Notation

For ease of reading, we standardize the notation used to describe SOTA.

- X: the set of all tokens within texts of interest. Each token is represented by a real vector $\mathbb{R}^\ell$ for some constant ℓ (this mapping from token to vector is called a *token embedding*).
- X': a set of sensitive tokens. We usually name sensitive tokens as x, x'. X' may be a sub- or super-set of X. For experiments, we use an initial set of sensitive tokens from X as seeds, which we then expand several-fold with additional tokens of a similar nature. For instance, if 'British' is a seed token, we may add 'French' and 'German', even if they were not initially in X.
- $y \in Y$: output of the token sanitization mechanism. Previous work differ in set Y; in the interest of fairness and ease of presentation, we set $Y = X'$ in all experiments.
- u: utility function; Δu: sensitivity of u (Exponential Mechanism Def. 3)
- $M : X' \to Y$: token sanitization mechanism

SanText [91] In SanText, the token sanitization mechanism $M : X' \to X'$ is based on the exponential mechanism (Def. 3), with a modification that the utility function sensitivity Δu is replaced by the constant 1 (see Thm. 1). The utility function is defined as $u(x, x') = -d(x, x')$, where $d(x, x')$ is a metric distance (e.g., Euclidean) between the real-vector embeddings of tokens x, x'. We formalize this mechanism in the Section 5.7. SanText achieves the following privacy guarantee.

Theorem 2 (Privacy of SanText)**.** *The token sanitization mechanism M of SanText, satisfies ϵ-MLDP.*

We note that SanText+ is a variation introduced in the same paper [91]. However, SanText+ sanitizes non-sensitive tokens as well as sensitive ones; thus, for fairness in utility comparisons (as sanitizing more tokens lowers the text utility), we focus on SanText.

CusText [21] CusText improves the utility of SanText by first partitioning all tokens in the lexicon X into disjoint sets called *clusters* based on token similarity. Then, given a fixed set of clusters, CusText performs exponential mechanism (Def. 3) within each cluster.

Since CusText only ever replaces x with tokens in the cluster containing x, CusText's privacy applies only within each cluster.

Theorem 3 (Privacy of CusText)**.** *Let C be a cluster. Let $M_C : C \to C$ be the mechanism M defined above, but with domain and range restricted to cluster C. Then M_C satisfies ϵ-LDP.*

However, CusText's mechanism M does not in general satisfy (metric) LDP, when there is more than one cluster. Intuitively, if tokens x, x' are in different clusters, $M(x)$ and $M(x')$ have disjoint supports and thus are easily distinguishable.

Theorem 4. *[Proof in Section 5.7] For any clustering with at least two clusters, the mechanism M defined in CusText cannot satisfy ϵ-(metric) LDP for any $\epsilon \in \mathbb{R}$.*

5.4 CluSanT: Cluster Exponential Mechanism with MLDP Guarantees

CluSanT first clusters tokens based on their similarity. It then sanitizes sensitive tokens by first selecting a cluster, then selecting the replacement token from within that cluster. This approach makes contextually relevant replacements, improving the utility of sanitized text over SanText while still maintaining MLDP. CluSanT's privacy guarantees hold for any clustering, allowing for flexible integration of different clustering methods.

In this section, we present two of the components in our CluSanT framework: cluster embedding (Sec. 5.4.1) and token sanitization (Sec. 5.4.2). The method of obtaining a token clustering is independent of this section and will be detailed in our experiments (see

Sec. 5.5). Through parameterizing our clustering, and cluster embedding with a parameter k, we obtain a spectrum of token sanitization mechanisms that range from SanText at one extreme and CusText at the other extreme (see Sec. 5.4.1). For now, we assume we already have a set of token clusters $\{C\}$.

Notation:

- Mapping f between token x and its vector representation in $\mathbb{R}^\ell$ is called a *token embedding*.
- Mapping f' between clusters and real vectors is called a *cluster embedding*.
- $\{C\}$: A clustering, a set of subsets C (*clusters*) partitioning $X' \cup Y$ (or X' if $Y = X'$). C_x is the (unique) cluster containing token x.
- $d_c : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$: Any distance function that is a metric. We extend this to measure the distance between clusters, i.e., $d_c(C, C') = d_c\left(f'(C),\ f'(C')\right)$ for clusters C, C'.
- $d : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$: Any distance measure (which is not assumed to be a metric, and may be different from d_c) between two tokens.

5.4.1 Cluster Embedding

We first define a cluster embedding given a token embedding and a clustering. Recall, a token embedding f is a mapping from a token to a real vector. A cluster embedding f', on the other hand, maps a cluster to a real vector. Our cluster embedding is parameterized by $k \geq 1$.

Our cluster embedding f' (Fig. 5.1) is parametrized by a standard token embedding f, a clustering, and k, which intuitively tunes how 'pushed apart' the clusters are from each other in the embedding. Looking ahead, our privacy Thm. 5 holds when the distance between clusters $d_c(C_x, C_{x'})$ can be increased by tuning k (the choice of d_c being a Lp-norm satisfies this).

Specifically, our cluster embedding f' embeds both tokens and clusters. It defines cluster embeddings by 'pushing' cluster apart by a factor of k (Step 1). Meanwhile, it maintains the original (according to f) difference between the embeddings of tokens from within the same cluster. This is done in Step 2(b), by adding the vector difference $(f(x) - C_x)$ to the new cluster centroid, $k \cdot C_x$.

How is cluster embedding used? Looking ahead, our sanitization mechanism: it first selects a cluster, then selects a token from within this cluster. By parametrizing f' with a larger k and a distance d_c that grows with k[4], we select more optimal clusters with higher probability.

Effect of parameter k. Since k only affects cluster selection (f'), we ensure that words within the same cluster remain 'indistinguishable' from each other, regardless of their embedding—we apply a standard LDP mechanism when selecting a token from within a cluster. Conversely, tokens from different clusters may be more distinguishable, depending on k. The larger k is, the more probability of choosing a better cluster, but the more distinguishable the clusters are (while still preserving MLDP). We formally describe the effect of k on privacy leakage in Section 5.7.3. Utility gains from larger k depend on evaluation metric; we give examples in our experiments (Sec. 5.5).

Defining a cluster-based embedding follows the spirit of [19] and [4]'s concept of geo-indistinguishability, where a radius naturally defines a cluster of close/indistinguishable geo-points within the radius. In our case, CluSanT forms clusters of words based on their (semantic/syntactic) similarity using word embeddings, where we leave the definition of 'similarity' up to user interpretation.

[4]Such as e.g., Euclidean used in SOTA.

Cluster Embedding f':

- **Inputs**: Parameter k, token embedding f, clustering $\{C\}$ of $X' \cup Y$
- **Output**: Embedding f'. On input a cluster C or token $x \in X' \cup Y$, f' outputs a real vector $\mathbb{R}^\ell$, which defines the embedding of the cluster C or token x.

1. For each cluster $C \in \{C\}$:
   (a) Compute the centroid of C. We overload notation and also use C to denote the centroid of this cluster. $C = \frac{1}{|C|} \sum_{x' \in C} f(x')$.
   (b) Define $f'(C) := k \cdot C$
2. For each token $x \in X' \cup Y$:
   (a) Compute the centroid of C_x. We overload notation and also use C_x to denote the centroid of this cluster. $C_x = \frac{1}{|C_x|} \sum_{x' \in C_x} f(x')$.
   (b) Define $f'(x) := k \cdot C_x + (f(x) - C_x)$

Figure 5.1: Cluster embedding with parameter k

Describing SanText and CusText in Terms of CluSanT

CluSanT can be parametrized, via the clustering, parameter k, and distances, to achieve a spectrum of ϵ-MLDP token sanitization mechanisms. We show that SanText and CusText are instantiations of CluSanT, situated at extremal ends of this parametrization.

Fact 1. *SanText and CusText are equivalent to CluSanT for specific choices of parameters.*

SanText. SanText's algorithm is the same as CluSanT (Fig. 5.2) parametrized by $k = 1$, d_c being Euclidean (same metric as SanText), and each cluster containing exactly one token (i.e., #clusters is equal to #tokens). Note this means that Step 1 is equivalent to SanText, and since each cluster has only one token, then Step 2 always chooses the same token (regardless of distance d), making this algorithm equivalent to SanText (recall, SanText does not consider token clustering).

CusText. CluSanT can be parametrized to *asymptotically* approach the behaviour of Cus-

Text, which always chooses a token from an 'optimal' cluster. We first define the clustering and distance d to be the same as CusText's, setting d_c as Euclidean, and letting $k \to \infty$. When $k \to \infty$, d_c between f' embeddings of different clusters is infinitely large, and Step 1 of Fig. 5.2 will with overwhelming probability choose the cluster C_x of the original token x.

5.4.2 Token Sanitization Mechanism for Metric LDP Guarantees

The main observation behind CluSanT's token sanitization mechanism is the following: CusText achieves good utility since it ensures that a token x is replaced only by another (possibly the same) token $x' \in C_x$, the cluster which x is in. However, by Thm. 4 we showed that this approach is impossible to achieve MLDP. Instead, CluSanT achieves privacy (and still good utility) by giving a *small* probability of selecting 'less good' clusters.

Our mechanism is in Fig. 5.2. Intuitively, Step 1 (cluster selection) is MLDP following the exponential mechanism-style approach of SanText. Then, Step 2 (selecting within a cluster) achieves guarantees similar to CusText. Parametrizing Steps 1 and 2 via the clustering, cluster embedding (k), and distances d, d_c, gives us a spectrum of ϵ-MLDP mechanisms that include SanText and CusText.

Theorem 5. *Consider d, d_c, f, f' (with parameter k), and the clustering are chosen s.t.:*

- *d_c is a metric.*
- *For all x, x' (using embedding f' for d_c, and f' for d): (1) $d_c(x, x') \geq 1$ or $d_c(x, x') \geq d(x, x')$, and (2) if $C_x \neq C_{x'}$, $d_c(C_x, C_{x'}) + 1 \leq 2 \cdot d_c(x, x')$.* [5]

Then, $M(x)$ in Fig. 5.2 achieves ϵ-metric LDP for metric d_c, and embedding f'.

Proof. Fix any $x, x' \in X', y \in Y$. If $x = x'$ then $\frac{\Pr(M(x)=y)}{\Pr(M(x')=y)} = 1 = e^0$ so the MLDP inequality trivially holds. Thus, consider $x \neq x'$.

[5] We refer to Section 5.7.2 for more details on satisfying these assumptions. In short, (1) can be satisfied with appropriate choices of distances and embeddings. (2) can be satisfied by choosing d_c based on any Lp-norm.

$M(x)$: The mechanism is parameterised by the set of clusters $\{C\}$ (where all tokens in clusters are in set $X' \cup Y$), token embedding f, cluster embedding f', metric d_c, distance d, and privacy parameter ϵ.
Input: token $x \in X'$
Output: token in Y

1. Choose a cluster C: Run exponential mechanism parametrised by $\epsilon_E = \epsilon/2$, input and output space are both $\{C\}$ (set of all clusters) with the cluster embedding f', utility $u_c(C, C') = -d_c(C, C')$, and setting parameter Δu_c to 1.
2. Choose token within cluster C: Run exponential mechanism parametrised by $\epsilon_E = \epsilon/2$, input space X', token embedding f, output space $C \cap Y$ (Y tokens in cluster C), utility $u(x, x') = -d(x, x')$, and setting $\Delta u = \max(1, \max_{x,x',y \in X'} |d(x, y) - d(x', y)|)$. (Note: Some embeddings, e.g., MPNet we use, are normalized and thus already have sensitivity 1).
3. Output the token chosen above.

Figure 5.2: CluSanT's ϵ-MLDP Sanitization Mechanism

We have $\Pr(M(x) = y)$ equal to $\Pr(M_1(x) = C_y) \Pr(M_2(x) = y | M_1(x) = C_y)$ where (1) $M_1(x) = C_y$ is the event that Step 1 of M chooses C_y, and (2) $M_2(x) = y | M_1(x) = C_y$ is the event that Step 2 of M' chooses token y, given Step 1 chooses C_y.

Conditioned on Step 1 choosing cluster C_y, Step 2 runs exponential mechanism with (both LDP and MLDP) privacy $\epsilon/2$. Thus, $\frac{\Pr(M_2(x)=y|M_1(x)=C_y)}{\Pr(M_2(x')=y|M_1(x')=C_y)} \leq \min\left(e^{\epsilon/2}, e^{\epsilon d(x,x')/2}\right)$. Moreover, by Thm 1,

$$\frac{\Pr(M_1(x) = C_y)}{\Pr(M_1(x') = C_y)} \leq \exp\left(\frac{\epsilon}{2}|d_c(C_x, C_y) - d_c(C_{x'}, C_y)|\right)$$

Thus,

$$\frac{\Pr(M(x) = y)}{\Pr(M(x') = y)} \leq e^{\frac{\epsilon}{2}|d_c(C_x,C_y) - d_c(C_{x'},C_y)|} e^{\frac{\epsilon}{2}\min(1, d(x,x'))}$$

Now consider the following two cases:

1. $C_x = C_{x'}$: Then, $|d_c(C_x, C_y) - d_c(C_{x'}, C_y)| = 0$ and $e^{\frac{\epsilon}{2}\min(1,d(x,x'))} \leq e^{\epsilon \cdot d_c(x,x')}$ by

theorem assumption (1) on d_c.

2. $C_x \neq C_{x'}$: Since d_c is a metric, $|d_c(C_x, C_y) - d_c(C_{x'}, C_y)| \leq d_c(C_x, C_{x'})$. Moreover, by assumption, for $C_x \neq C'_x$, $d_c(C_x, C_{x'}) + 1 \leq 2 \cdot d_c(x, x')$. Thus,

$$\begin{aligned} e^{\frac{\epsilon}{2}(|d_c(C_x,C_y)-d_c(C_{x'},C_y)|+1)} &\leq e^{\frac{\epsilon}{2}(2\cdot d_c(x,x'))} \\ &= e^{\epsilon \cdot d_c(x,x')} \end{aligned}$$

5.5 Experiments

Previous work [44, 91, 21] evaluated the quality of sanitized text based on the performance of downstream tasks (e.g., sentiment analysis) on sanitized text. In this work, we evaluate the quality of sanitized text using more direct metrics that capture the semantic integrity and linguistic naturalness of sanitized text.

Metrics and Dataset We evaluate sanitized text primarily with *semantic similarity* and *perplexity*, and four additional metrics assessing common sense, coherence, cohesiveness, and grammar quality, using GPT-4o. In Section 5.7.4 we show (1) example sanitized texts, evaluated using cosine similarity, and (2) evaluations of a sanitized text validation set based on the SST2 [78] dataset used in SanText and CusText.

Cosine/Semantic similarity is measured using embedding vectors from the all-MiniLM-L6-v2 (sentence embedder) model[6]. We compute cosine similarity, a fundamental component of many downstream text mining tasks such as sentiment classification [82], between embeddings of original and sanitized texts to assess semantic preservation.

Perplexity measures the naturalness of the sanitized text by how well it aligns with typical language patterns, with lower perplexity indicating more natural text (higher probability).

[6] `https://docs.trychroma.com/guides/embeddings`

We evaluate the perplexity of sanitized texts with GPT-2.

GPT-4o is used to evaluate grammar, common sense, coherence, and cohesiveness. LLMs' capabilities in assessing these metrics has been shown to match or surpass human evaluators in accuracy and consistency [23, 90] in various NLP evaluations (e.g., RAG).

We use the TAB dataset [71], which includes 1,268 annotated English-language court cases from the European Court of Human Rights. This dataset offers a robust framework for evaluating general-purpose text anonymization, with high-quality annotations and diverse content. More experiment setup details are in Section 5.7.4.

Experimental Methodology In our experiments [7], we use a simple CusText clustering method [21]: given a set of tokens X to cluster, it randomly picks a token x, creates a cluster C_x, and inserts into C_x the top $h-1$ tokens similar to x from X, simultaneously removing them from X. This process is repeated until X is empty, resulting in each cluster containing exactly h tokens. While this simple clustering method can be improved, the choice of clustering method is orthogonal to our work. By varying h, we can control the size and number of clusters. In our study, we test with 40, 180, 360, and 720 clusters.

Augmented Token Set We improve previous approaches like SanText and CusText by augmenting the set of sensitive words and phrases, making it more realistic and contextual. We consider the set X' as all sensitive words or phrases from the TAB dataset, supplemented with 100 words/phrases of similar nature for each using GPT-4o. For example, for "Sinn Fein headquarters," we obtained phrases of similar nature like "Labour Party headquarters," "Conservative Party headquarters," etc., rather than only similar terms like "Irish" which, while similar in vector embedding space, are not of the same nature. This approach extends the set of sensitive tokens to include additional, realistic phrases not orig-

[7] For code and more details on our experiments please see `https://github.com/AwonSomeSauce/CluSanT.git`.

inally in a text collection but still sensitive. In contrast, SanText and CusText recognizing the limitations of a restricted set of sensitive tokens, attempt to mitigate this by allowing replacements with non-sensitive words. However, this method often leads to replacements that do not always make sense. For example, replacing "Sinn Fein headquarters" with a non-sensitive word, such as "Irish" can render the text nonsensical.

Multi-word Embeddings SanText and CusText rely on single-word embeddings like GloVe [70], which cannot directly handle multi-word phrases such as "Sinn Fein headquarters." Our approach, on the other hand, employs the all-MiniLM-L6-v2 sentence embedder, designed to handle phrases and provide more accurate contextual representations. Our experiments ensure fairness for SanText and CusText by using the same set X' of words/phrases for replacement and the same token embedder, all-MiniLM-L6-v2. Finally, we use Euclidean distance for all methods.

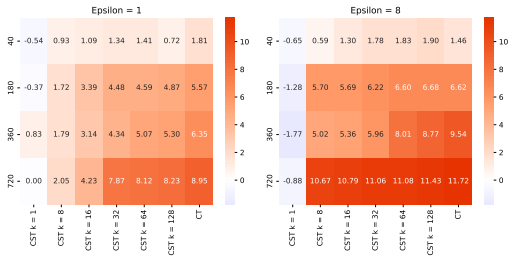


Figure 5.3: Semantic similarity improvement over SanText (%). CluSanT abbreviated by CST, CusText by CT. Horizontal axis varies parameter k of CluSanT. Vertical axis varies the number of clusters. Same axes apply for the other heatmaps as well. Unless otherwise mentioned, the higher, the better.

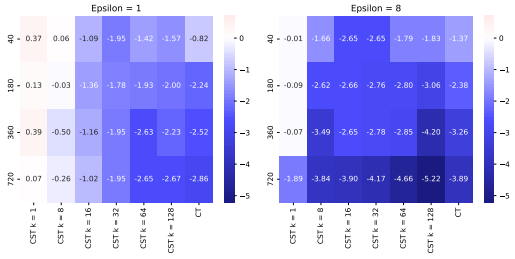


Figure 5.4: Peplexity improvement over SanText (%); the lower, the better

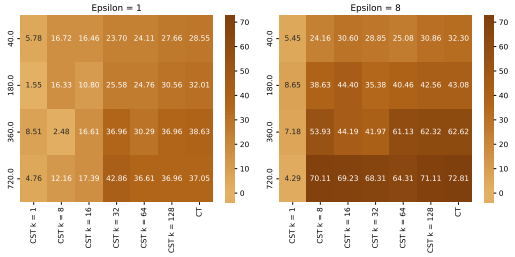


Figure 5.5: Common sense improv. over SanText (%)

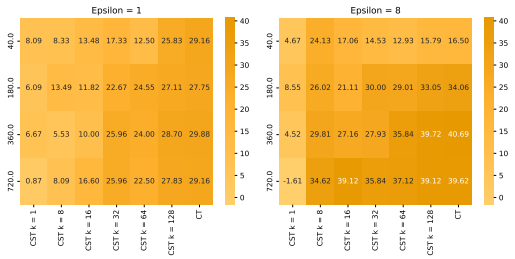


Figure 5.6: Coherence improvement over SanText (%)

Numerical Results We show partial experimental results, in Figures 5.3, 5.4, 5.5, 5.6 and full results in Figures 5.7, 5.8, 5.9, 5.10, 5.11, 5.12 in Section 5.7. Figures show improvements achieved by CluSanT in terms of semantic similarity, perplexity, common sense, coherence, cohesiveness, and grammar over SanText. For all metrics except perplexity, higher scores are better, while for perplexity, lower scores are preferred. We abbreviate CluSanT by CST and CusText by CT. We consider CT as a special version of CluSanT where $k = \infty$.

We plotted the number of clusters on the vertical axis and the centroid pushing factor k of CluSanT on the horizontal axis, creating heatmaps for each ϵ value considered: 0.5, 1, 2, 4, 8, 16. Due to space constraints, we only show results for $\epsilon = 1$ and $\epsilon = 8$ here and include the rest in the Section 5.7.

For all ϵ values, as k increases (moving right in the maps), the semantic similarity improvement (over SanText) increases. Additionally, as the number of clusters increases (moving down), semantic similarity improvement also increases.

The most significant improvement of CluSanT over SanText is observed for $\epsilon = 8$ and

720 clusters. Generally, the more clusters used, the greater the improvement over SanText. For $\epsilon = 16$, the improvement in semantic similarity for CluSanT over SanText is not as pronounced as for $\epsilon = 8$. This is because, while the semantic similarity of sanitized text to the original text for CluSanT approaches 1 for this ϵ (the highest value possible, being a cosine similarity), the semantic similarity for SanText also increases for larger ϵ values, resulting in a slightly reduced improvement margin.

Similar trends are observed for perplexity, common sense, and coherence, as well as for grammar and cohesiveness (detailed in Section 5.7). As the number of clusters and k increase, CluSanT's performance improves significantly over SanText, approaching that of CusText. While CusText shows better performance across metrics, it is only marginally better than CluSanT. However, this comes at the cost of weaker privacy guarantees. We note that these metrics represent general trends; as LLM judgments can be noisy, smaller k values may occasionally yield better results.

5.6 Conclusions

In this chapter, we addressed the challenge of privacy-preserving text sanitization in Natural Language Processing (NLP), focusing on how to apply Differential Privacy (DP) without sacrificing the semantic utility and linguistic quality of the text. Existing approaches—such as SanText and CusText—represent two extremes in the privacy-utility trade-off: SanText offers strong privacy but often yields degraded, low-utility outputs, while CusText improves text quality at the expense of formal privacy guarantees.

To bridge this gap, we introduced CluSanT, a novel framework for text sanitization that provides formal Metric Local Differential Privacy (MLDP) guarantees while maintaining high semantic and linguistic fidelity. The key innovation in CluSanT is its use of a parameterized, two-step sanitization process that separates token clustering from cluster selection,

allowing for fine-grained control over the privacy-utility balance. Our framework generalizes and subsumes existing approaches: SanText and CusText appear as special cases of CluSanT under particular parameter settings, establishing CluSanT as a superset in the design space of DP-based text sanitization.

Through a series of technical innovations—including a cluster-based exponential mechanism, a flexible amplification parameter k, and multi-word embeddings—we addressed the key limitations of prior work. Importantly, CluSanT 's privacy guarantees are agnostic to the specific clustering algorithm used, making it extensible and adaptable to different domains, languages, or data types. This modular design encourages further research into optimizing cluster quality to improve semantic preservation while preserving privacy guarantees.

To evaluate CluSanT, we employed a set of direct and interpretable metrics that assess both semantic similarity and linguistic naturalness, including perplexity, common sense, coherence, grammar, and cohesiveness. These were measured using both sentence-level embeddings and LLM-based evaluation (GPT-4o), providing a more realistic assessment than downstream proxy tasks like sentiment classification. Experimental results on the TAB dataset demonstrate that CluSanT consistently outperforms SanText across all metrics and closely approaches the utility levels of CusText—all while maintaining strong DP guarantees. We further showed how the choice of parameter k and number of clusters enables smooth trade-offs between utility and privacy, giving practitioners the ability to tune behavior for specific application needs.

From a broader perspective, this work reinforces a key theme of the thesis: embedding-based representations can serve not only as tools for understanding bias or modeling network dynamics but also as mechanisms for protecting privacy. In CluSanT, vector embeddings were leveraged not just for utility, but for regulating privacy-preserving replacements in a way that is both mathematically rigorous and linguistically meaningful.

Looking forward, several promising directions remain. Improving the quality and stability of clustering algorithms, adapting CluSanT to streaming or multilingual settings, and incorporating user feedback into replacement selection are all avenues for advancing this framework. Additionally, as LLMs themselves evolve, the utility-privacy interface will likely shift, making it vital to periodically reassess these mechanisms in light of changing model behavior and societal expectations.

In conclusion, CluSanT offers a principled, flexible, and empirically validated approach to text sanitization, providing a viable foundation for privacy-aware NLP applications that do not compromise on utility or readability.

5.7 Additional Details and Results

5.7.1 SanText and CusText

SanText

Below, we detail the operational steps of the token sanitization mechanism M of SanText when processing an input token x:

1. M is parameterized by the privacy parameter ϵ and employs a metric d to measure distances between tokens.
2. The utility function u is defined such that $u(x, x) = -d(x, x)$. Under this definition, M selects an output from Y based on the exponential mechanism (Def. 3) tailored for the specified ϵ, but with parameter Δu set as 1.

CusText

We describe CusText's token sanitization algorithm $M : X' \to X$ for input token x.

1. M is parametrized by the formed clusters, a distance d, and the privacy parameter ϵ.

2. Let $C \subseteq X$ be the cluster x belongs in.

3. Let utility $u : C \times C \to \mathbb{R}$ be the negative of the normalised distance $u(x, y) = -\frac{d(x,y)-d_{\min}}{d_{\max}-d_{\min}}$ ($d_{\min} = \min_{x,y \in C} d(x, y)$ and $d_{\max} = \max_{x,y \in C} d(x, y)$), so that sensitivity $\Delta u = 1$.

4. Using the above utility function, replace x with some token in C via the exponential mechanism (Def. 3) for privacy ϵ.

We present the proof for Thm. 4 that CusText cannot achieve standard (M)LDP when there is more than one cluster.

Proof. We prove first for LDP. Suppose for contradiction that there exists $\epsilon \in \mathbb{R}$ such that M satisfies ϵ-LDP. Then for all $x, x' \in X', y \in X$, this inequality must hold:

$$\Pr(M(x) = y) \leq e^{\epsilon} \Pr(M(x') = y)$$

Since there are at least two clusters, there must exist $x, x' \in X'$ that belong in different clusters. Let y be a token in the cluster of x, which means y is not in the cluster of x'. This means that $\Pr(M(x) = y) > 0$ and $e^{\epsilon} \Pr(M(x') = y) = e^{\epsilon} \cdot 0 = 0$. Thus the above inequality cannot hold, which is a contradiction, and thus there is no ϵ for which M is ϵ-LDP.

The proof for metric LDP follows since if M is ϵ-metric LDP then M is $\epsilon \cdot \Delta u$-LDP. Since the lexicon is finite, Δu maximises over a finite set and is also finite. Thus, if M is ϵ-metric LDP then it is ϵ'-LDP for some finite ϵ' (which we just proved is impossible). □

5.7.2 Setting Parameters Satisfying Theorem Assumptions

We discuss how to parameterize CluSanT in order to leverage our general privacy Theorem 5. We note that while we give specific examples of parameters below (e.g., d_c set as

Euclidean), our theorem assumptions are stated more generally and may be satisfied via other instantiations.

Assumption (1) can be satisfied by an appropriate setting of the embedding or distances d_c, d. For example, one can choose d_c as Euclidean, and setting k to be large enough so that $d_c(x, x') \geq d(x, x')$ (for embedding f' for d_c and embedding f for d; recall k does not change f). Another way is to achieve $d_c(x, x') \geq 1$ for $x \neq x'$, by normalizing embeddings.

Fact 2. ***Assumption (2)*** *can be satisfied for d_c being any Lp-norm (e.g., Euclidean/L2-norm used in SanText), and a large enough k.*

Proof. Assume tokens below use embedding parameterised by k. By triangle inequality (since d_c is a metric), we can write

$$d_c(x, x') \geq d_c(C_x, C_{x'}) - d_c(x, C_x) - d_c(x', C_{x'})$$

Now multiply both sides by 2:

$$\begin{aligned}
&2 \cdot d_c(x, x') \\
&\geq 2 \cdot d_c(C_x, C_{x'}) - 2 \cdot (d_c(x, C_x) - d_c(x', C_{x'})) \\
&= d_c(C_x, C_{x'}) + [d_c(C_x, C_{x'}) \\
&\quad - 2 \cdot (d_c(x, C_x) - d_c(x', C_{x'}))]
\end{aligned}$$

To prove our inequality, we just need that the above is $\geq d_c(C_x, C_{x'}) + 1$. We note that we already have the "$d_c(C_x, C_{x'})$" part of the sum, so we want

$$d_c(C_x, C_{x'}) - 2 \cdot (d_c(x, C_x) - d_c(x', C_{x'})) \geq 1$$

For $d_c(y, z)$ equal to the Lp-norm of $y - z$, $\lim_{k \to \infty} d_c(x, x') \to \infty$, but $(d_c(x, C_x) -$

$d_c(x', C_{x'}))$ remains constant (since distance between x and C_x is unaffected by k). Thus, we can always find a k such that the above inequality holds.

□

Intuition for the above proof: Recall in CluSanT we write the embedding of point x in terms of the cluster centroid $k \cdot C_x$ when using embedding f parametrised by k. So

$$x = k \cdot C_x + (C_x - x)$$

and

$$x' = k \cdot C_{x'} + (C_{x'} - x')$$

Importantly, the "distance to the centroid", $(C_x - x)$ or $(C_{x'} - x')$ remains constant regardless of k. This is true for distances based on Lp-norm, like Euclidean used in the SOTA. So when k is large, this "$(C_x - x)$" term becomes less significant, so approximately,

$$\begin{aligned} & 2 \cdot d_c(x, x') \\ & \approx 2 \cdot d_c(k \cdot C_x, k \cdot C_{x'}) \\ & = 2 \cdot d_c(f(C_x), f(C_{x'})) \text{ for our embedding } f \text{ parametrised by } k \end{aligned}$$

for some large enough k, the above is $\geq d_c(f(C_x), f(C_{x'})) + 1$

5.7.3 Effect of Parameter k on Privacy

We formally quantify the effect privacy leakage of CluSanT based on k, for the example where d_c is any Lp-norm (i.e., $d_c(x, x') = ||x - x'||_p$. Informally, k linearly degrades LDP guarantees.

Fact 3. *Consider an instantiation M of CluSanT that satisfies the assumptions of Thm. 5, and let d_c be any Lp-norm. Then this instantiation satisfies $\epsilon\Delta$-LDP, where $\Delta = \max_{x,x'} d_c(x, x')+$*

$k \cdot \max_{C_x, C_{x'}} d_c(C_x, C_{x'})$ *(using embedding f).*

Proof. M satisfies ϵ-MLDP by Thm. 5, that is, $\forall x, x', y$ (We explicitly show the embedding here for clarity.):

$$\begin{aligned}\Pr(M(x) = y) &\leq e^{\epsilon \cdot d_c(f'(x), f'(x'))} \Pr(M(x') = y) \\ &= e^{\epsilon ||f'(x) - f'(x')||_p} \Pr(M(x') = y) \\ &\leq e^{\epsilon \max_{x,x'} ||f'(x) - f'(x')||_p} \Pr(M(x') = y)\end{aligned}$$

Here,

$$\begin{aligned}&||f'(x) - f'(x')||_p \\ &= ||f(x) + k \cdot f(C_x) - (f(x') + k \cdot f(C_{x'}))||_p \\ &\leq ||f(x) - f(x')||_p + ||k \cdot C_x - k \cdot C_{x'}||_p\end{aligned}$$

The first inequality above is due to d_c being a metric. Thus,

$$\begin{aligned}&\max_{x,x'} ||f'(x) - f'(x')||_p \\ &\leq \max_{x,x'} d_c(f(x), f(x')) + k \cdot \max_{C_x, C_{x'}} d_c(f(C_x), f(C_{x'}))\end{aligned}$$

□

5.7.4 Detailed Experimental Results

Here we give the detailed prompt for obtaining the extended set X' of sensitive tokens using GPT-4o.

> *If I give you a word or phrase, example "southern norrland," can you give me 100 similar words/phrases of the same category?*
>
> *For example:*

- *If it is a location, give me other locations that are similar in nature.*
- *If it is an organization, give me other organizations that are similar.*
- *If it is an object, give me other objects that are similar.*

The similarity should be in terms of the category and characteristics of the entity. The words you give should make sense if used as a replacement for the original word/phrase in a similar context.

Format output as a list of words/phrases:

[*word/phrase1, word/phrase2, ...*]

Here the context that "{*search_phrase*}*" was used in: "*{*context*}*".*

For example, when the search phrase was 'sarpsborg city court (tingrett)' with its context in the TAB text, the output we received from GPT-4o was:

['oslo district court', 'bergen district court', 'trondheim district court', 'stavanger district court', 'kristiansand district court', 'tromsø district court', 'drammen district court', 'fredrikstad district court', 'skien district court', 'ålesund district court', 'bodø district court', 'hamar district court', 'molde district court', 'haugesund district court', etc].

Now we give detailed experimental results on semantic similarity (Figure 5.7), perplexity (Figure 5.8), grammar (Figure 5.9), common sense (Figure 5.10), coherence (Figure 5.11), and cohesiveness (Figure 5.12). For perplexity we used GPT-2. To judge the grammar, common sense, coherence, and cohesiveness of santized text, we used GPT-4o, with the following prompt.

Could you please evaluate the following passage for its grammar, common sense, coherence, and cohesiveness? Score it on a scale from 1 to 5, where 1 is the lowest (poor quality) and 5 is the highest (excellent quality).

You should score based on these criteria:

- ***Grammar**: Are the sentences structured correctly?*
- ***Common sense**: Does the content make logical sense in the real world?*
- ***Coherence**: Do the ideas flow logically from one sentence to another?*
- ***Cohesiveness**: Do all parts of the text come together in a unified whole?*

Please ONLY respond in JSON format with the four keys 'grammar', 'common sense', 'coherence', and 'cohesiveness', each with a score attached to them.

CluSanT's improvement over SanText generally increases with the number of clusters used. Increasing the number of clusters and parameter k significantly enhances CluSanT 's performance, approaching that of CusText. Although CusText performs better across metrics, it offers weaker privacy guarantees. Note that these metrics represent general trends, and due to noisy judgments from LLMs, smaller k values can sometimes yield better results.

In the following, we present an example of an original text from the court dataset collected by [71], along with the substitutions made by SanText, CluSanT with the number of clusters 100 and $k = 16, 64$, and CusText. We also give the cosine similarities for each substitution and average similarities.

The original text is as follows:

> The case originated in an application (no. 18308/02) against the Republic of Turkey lodged with the Court under Article 34 of the Convention for the Protection of Human Rights and Fundamental Freedoms ("the Convention") by two Turkish nationals.

We do not show the whole original text because even though it is public, it still contains sensitive information.

Comparison of Substitution Methods

SanText ($\epsilon = 4$)

- *Aliağa Public Prosecutor → 60,000 norwegian kroner (nok) (approximately 7,500 euros)*: 0.0633
- *Aliağa Criminal Court → district court of öland*: 0.5709
- *Court of Cassation → court of the township*: 0.5470
- *Republic of Turkey → legal and services office*: 0.2239
- *Turkish Government → pdki (democratic party of iranian kurdistan)*: 0.5242

Average Cosine Similarity: 0.2264

CluSanT ($\epsilon = 4$, clusters = 1000, $k = 16$)

- *Aliağa Public Prosecutor → manisa high criminal court*: 0.5344
- *Aliağa Criminal Court → urban planning court*: 0.5415
- *Court of Cassation → court of the vicar-general*: 0.6089
- *Republic of Turkey → supreme court of north macedonia*: 0.3797
- *Turkish Government → halkın gücü (people's power)*: 0.6460

Average Cosine Similarity: 0.5718

CluSanT ($\epsilon = 4$, clusters = 1000, $k = 64$)

- *Aliağa Public Prosecutor → bialya lead prosecutor*: 0.6840
- *Aliağa Criminal Court → elblag regional court*: 0.6085
- *Court of Cassation → court of the steward of the marshalsea*: 0.5849

- *Republic of Turkey* $\rightarrow$ *republic of slovenia*: 0.6537
- *Turkish Government* $\rightarrow$ *balıkesir, edremit*: 0.5269

Average Cosine Similarity: 0.6985

CusText ($\epsilon = 4$, clusters = 1000, equivalent to $k \rightarrow \infty$)

- *Aliağa Public Prosecutor* $\rightarrow$ *turhal public prosecutor*: 0.6318
- *Aliağa Criminal Court* $\rightarrow$ *storfors district court (storfors tingsrätt)*: 0.5465
- *Court of Cassation* $\rightarrow$ *court of the lord high admiral*: 0.5305
- *Republic of Turkey* $\rightarrow$ *republic of azerbaijan*: 0.7156
- *Turkish Government* $\rightarrow$ *ottoman empire*: 0.5591

Average Cosine Similarity: 0.6703

SanText produces some substitutions that are quite off, for instance, replacing "Aliağa Public Prosecutor" with "60,000 Norwegian kroner (NOK) (approximately 7,500 euros)", which is meaningless in this context.

CluSanT with $k = 16$ provides more contextually appropriate substitutions compared to SanText. However, improvements are seen with a higher k value. **CluSanT** with parameters $(\epsilon = 4, \text{clusters} = 1000, k = 64)$ provides the most contextually appropriate substitutions with the highest average cosine similarity, making it the best choice for preserving the meaning and context of the original text.

CusText also provides reasonable substitutions but occasionally diverges, such as replacing "Turkish Government" with "Ottoman Empire."

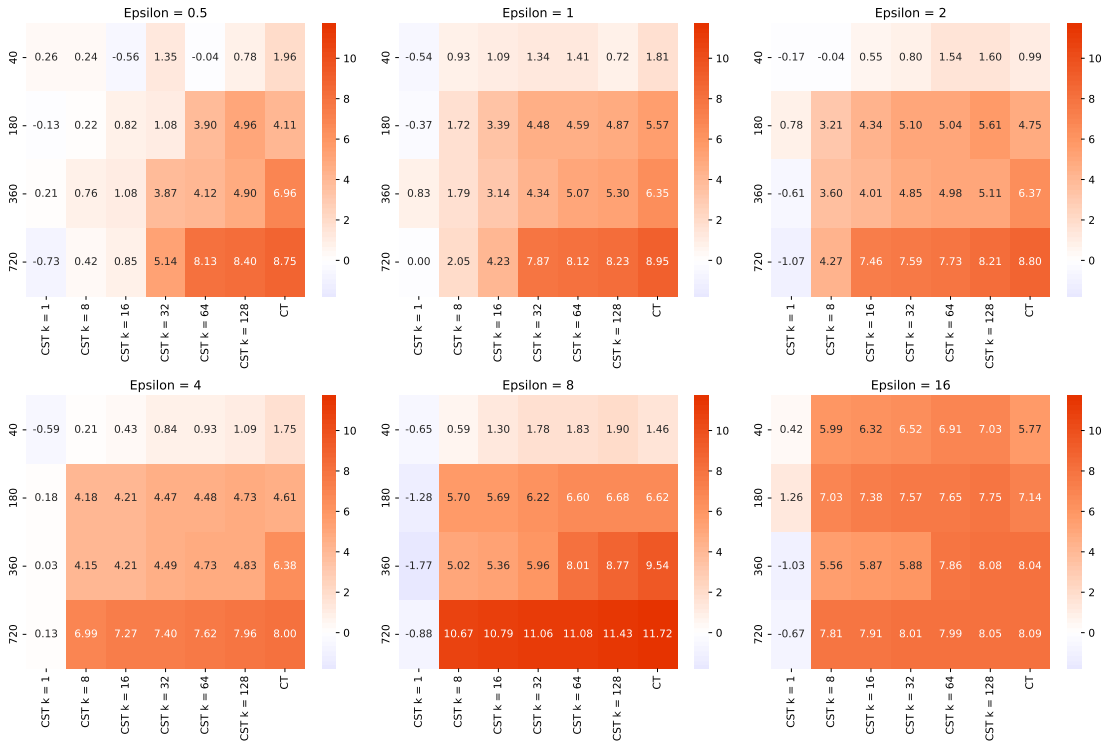


Figure 5.7: Semantic similarity improvement over SanText (%); the higher, the better. CluSanT abbr. by CST and CusText by CT. Horizontal axis varies parameter k of CluSanT. Vertical axis varies the number of clusters. Same axes apply for the other heatmaps as well.

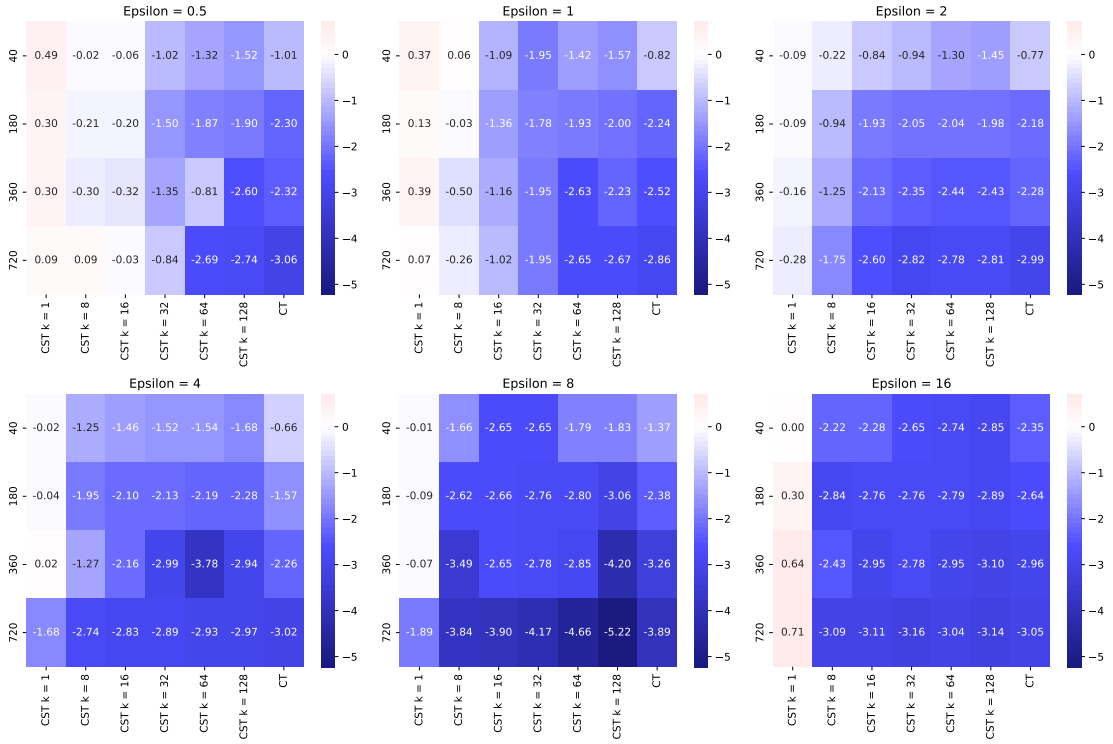


Figure 5.8: Peplexity improvement over SanText (%); the lower, the better.

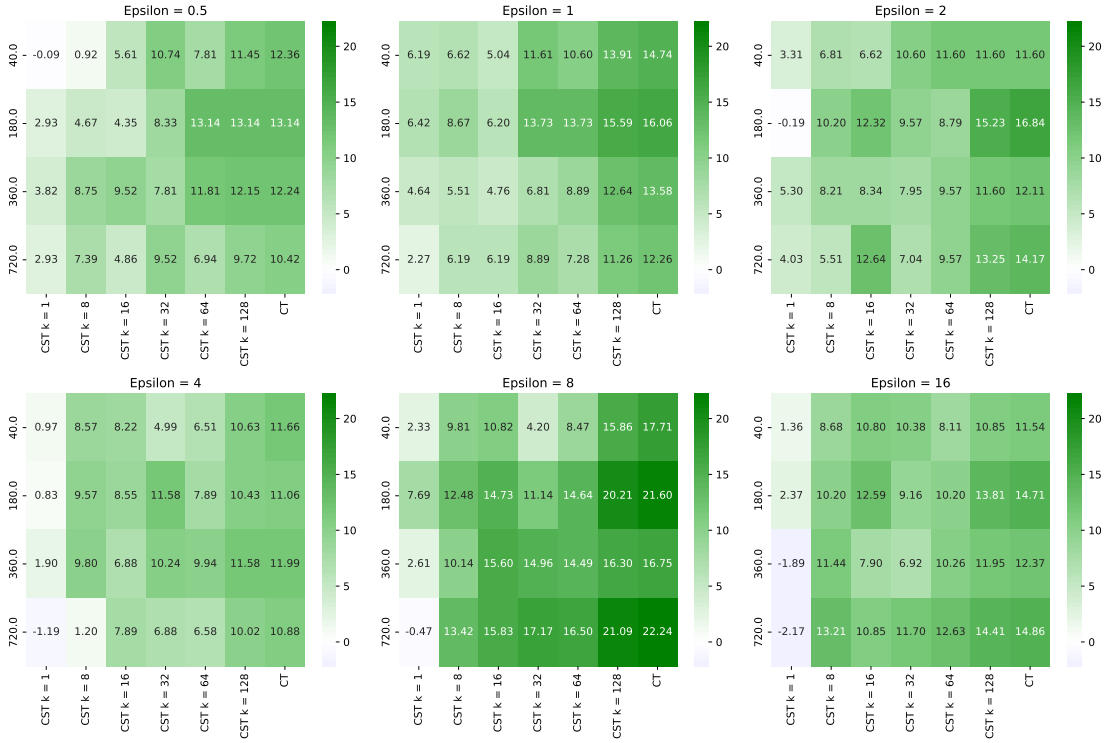


Figure 5.9: Grammar improvement over SanText (%); the higher, the better.

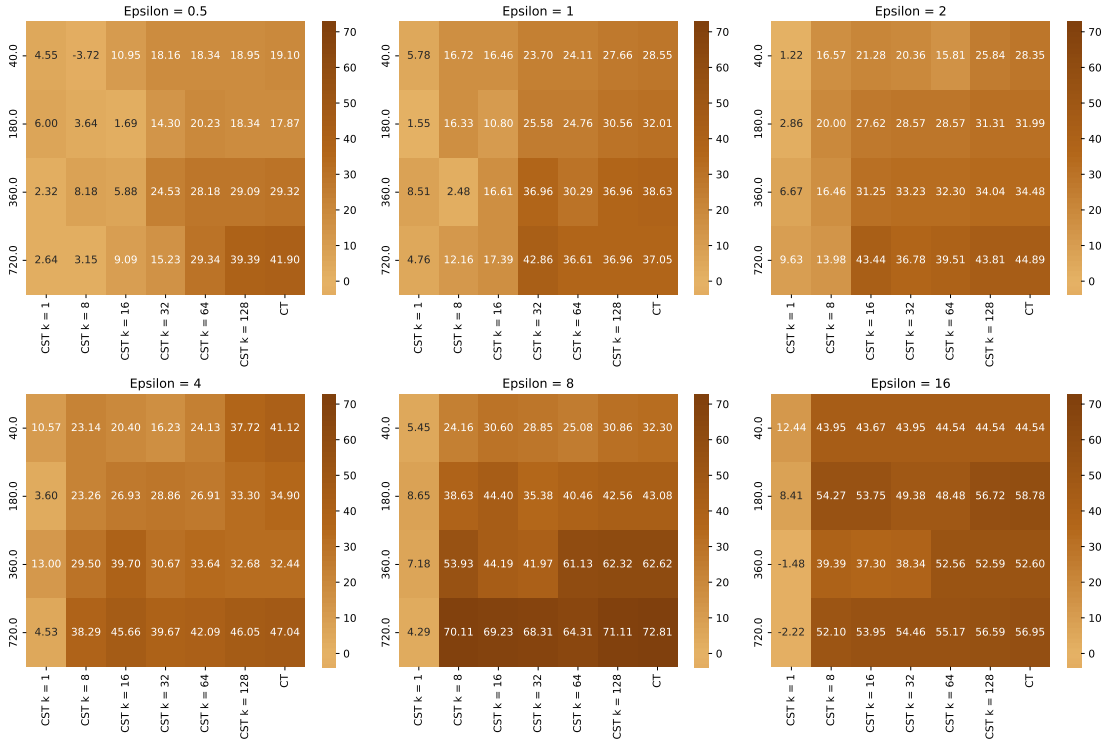


Figure 5.10: Common Sense improvement over SanText (%); the higher, the better.

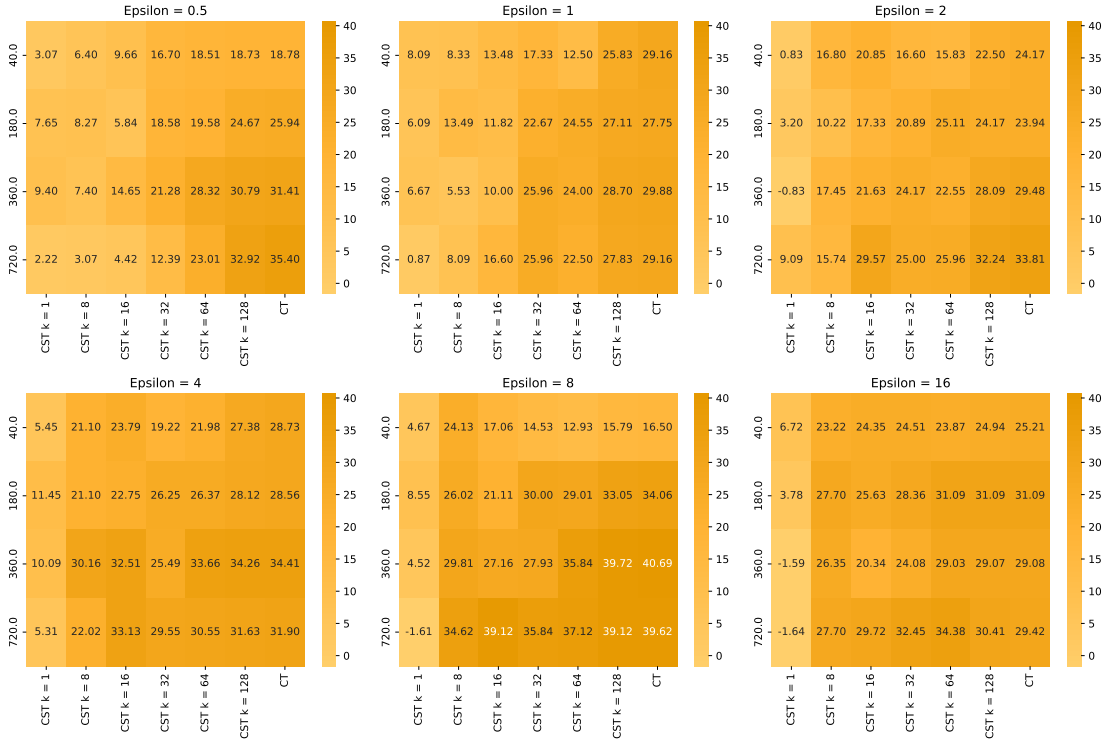


Figure 5.11: Coherence improvement over SanText (%); the higher, the better.

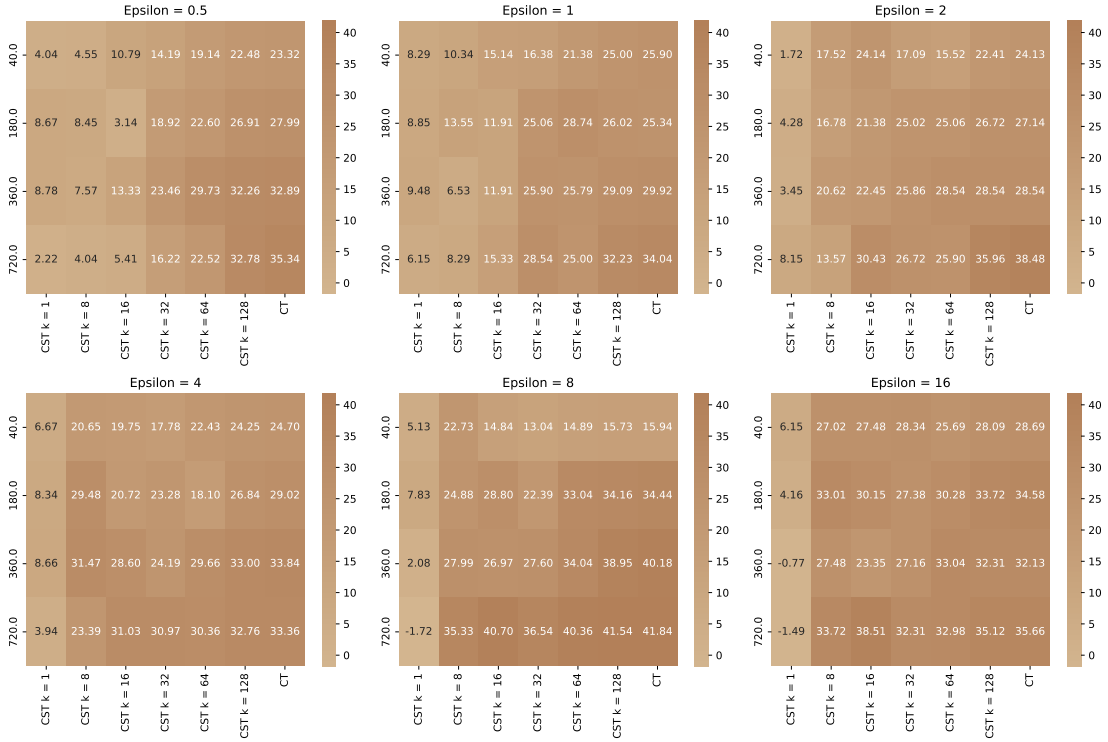


Figure 5.12: Cohesiveness improvement over SanText (%); the higher, the better.

SST2: Binary Classification for Sentiment Analysis

We further demonstrate how CluSanT can be used to improve the utility in downstream tasks through the SST2 dataset [78] which has also been used to evaluate SanText, CusText (though their experiments focused on sanitizing text for training). For this experiment (Table 5.1), we evaluated an already-trained model for the task Binary Classification for Sentiment Analysis on a validation set sanitized either through SanText, CusText, or CluSanT (various k parameters). We see that with higher k, we achieve significantly higher accuracy and lower loss than SanText and approaching that of CusText (while still achieving standard MLDP guarantees).

ϵ	num clusters	Mechanism	k	Accuracy	Loss
N/A	N/A	Unsanitized	N/A	0.91954	0.263152
1	N/A	Santext	N/A	0.678161	1.467184
1	336	CluSanT	8	0.643678	1.762656
1	336	CluSanT	16	0.666667	1.544436
1	336	CluSanT	32	0.724138	1.129536
1	336	Custext	N/A	0.804598	0.828578
4	N/A	Santext	N/A	0.62069	1.799631
4	336	CluSanT	1	0.666667	1.569011
4	336	CluSanT	8	0.712644	1.466076
4	336	CluSanT	16	0.735632	1.144738
4	336	CluSanT	32	0.793103	1.056569
4	336	Custext	N/A	0.724138	1.346923
8	N/A	Santext	N/A	0.703561	1.394245
8	336	CluSanT	1	0.678161	1.420536
8	336	CluSanT	8	0.689655	1.5434
8	336	CluSanT	16	0.770115	0.993965
8	336	CluSanT	32	0.793103	0.860854
8	336	Custext	N/A	0.827586	0.760091
16	N/A	Santext	N/A	0.712644	1.474932
16	336	CluSanT	1	0.804598	1.168481
16	336	CluSanT	8	0.850575	0.564254
16	336	CluSanT	16	0.885057	0.463717
16	336	CluSanT	32	0.882357	0.463717
16	336	Custext	N/A	0.873563	0.486622

Table 5.1: SST2 Binary Classification for Text Sentiment Analysis [78] on existing trained model when validation set is sanitized with various mechanisms.

Chapter 6

Epilogue

This thesis set out to explore three critical dimensions of responsible AI—fairness, bias, and privacy—through the lens of vector-based representations. Across distinct yet interconnected domains, we demonstrated how embedding-driven systems not only enable powerful AI capabilities but also encode, propagate, and potentially mitigate structural inequities. The result is a unified perspective that brings together technical rigor and ethical awareness in addressing some of the most pressing concerns in modern machine learning.

In the first part, we examined algorithmic fairness in social recommender systems, focusing on the long-term visibility of minority groups within evolving networks. By shifting the lens from attribute-based to structure-based community definitions, we revealed how standard algorithms often marginalize organically emerging minority clusters. Our work introduced `MinWalk`, a fairness-aware recommendation algorithm that balances minority visibility with network stability. Through extensive simulation on real-world networks, we showed how this approach outperforms traditional baselines while avoiding the risks of overcompensation and algorithmic backlash. This work advances the conversation on fairness from static metrics to dynamic, user-sensitive network interventions.

In the second part, we turned to demographic bias in Large Language Models (LLMs),

starting from representational disparities in word embeddings and extending to biased behavior in downstream applications. Our analysis of modern LLMs revealed persistent gender and race-based biases, both in semantic association and in application contexts such as consumer product recommendations. By applying clustering, visualization, and classification techniques, we provided a quantitative and thematic understanding of how stereotypes are encoded and reproduced. Our findings highlighted how surface-level outputs reflect deeper biases embedded in learned representations—emphasizing the ongoing need for interpretability, accountability, and auditability in language technologies.

In the final part, we addressed privacy in NLP, proposing a novel framework, CluSanT, for Differentially Private text sanitization. Recognizing the limitations of existing methods that either degrade text utility or fail to guarantee robust privacy, we designed a flexible mechanism that combines token clustering, cluster embeddings, and a parameterized MLDP algorithm. Our experimental evaluations—leveraging both semantic similarity and LLM-based metrics—demonstrated that CluSanT offers a compelling balance, matching or surpassing prior work in utility while maintaining strong formal privacy guarantees. This work not only contributes a practical tool for anonymization but also establishes a path forward for building privacy-preserving NLP systems that retain human readability and coherence.

Across these three pillars—fairness, bias, and privacy—a common thread emerges: representation matters. Whether modeling users in a graph, concepts in a vector space, or tokens in a text stream, how we represent information fundamentally shapes algorithmic outcomes. Embeddings are not just mathematical tools—they are sociotechnical artifacts that carry implications for who is seen, how they are framed, and what remains hidden. By interrogating and refining these representations, this thesis contributes to a growing body of work that seeks to build AI systems that are not only intelligent but also just, inclusive, and respectful of user rights.

Looking ahead, the field faces an urgent mandate to move beyond abstract performance metrics and consider the societal stakes of AI deployment. Algorithmic fairness must grapple with dynamic, multi-agent environments. Bias audits must evolve alongside increasingly complex and opaque models. Privacy must become more than a compliance checkbox—it must be a design principle. To meet these challenges, future work must bridge technical depth with normative insight, embedding ethics into the core of model development, evaluation, and deployment.

In this spirit, this thesis offers not only a collection of contributions but also a framework for thinking about responsible AI: one that treats fairness, bias, and privacy not as isolated concerns, but as intertwined facets of how we design systems, make decisions, and shape digital life.

Bibliography

[1] Mohamed Abdalla and Moustafa Abdalla. The grey hoodie project: Big tobacco, big tech, and the threat on academic integrity. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 287–297, 2021.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.

[3] Mário Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Anna Pazii. Local differential privacy on metric spaces: optimizing the trade-off with utility. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 262–267. IEEE, 2018.

[4] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914, 2013.

[5] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*, 2021.

[6] Eytan Bakshy, Solomon Messing, and Lada A. Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015.

[7] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[8] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[9] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[10] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in Neural Information Processing Systems*, 29:4349–4357, 2016.

[11] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[12] Stephen P Borgatti and Daniel S Halgin. Analyzing social networks. In *SAGE Publications Sage UK: London, England*, 2011.

[13] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[14] Aylin Caliskan, Parth Ajay Pimparkar, Tessa Charlesworth, Robert Wolfe, and Mahzarin R. Banaji. Gender bias in word embeddings: A comprehensive analysis

of frequency, syntax, and semantics. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 156–170, 2022.

[15] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[16] Ricardo Silva Carvalho, Theodore Vasiloudis, Oluwaseyi Feyisetan, and Ke Wang. Tem: High utility metric differential privacy on text. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 883–890. SIAM, 2023.

[17] Òscar Celma and Pedro Herrera. A new approach to evaluating novel recommendations. In *SIGIR*, pages 379–386. ACM, 2010.

[18] Kaytlin Chaloner and Alfredo Maldonado. Measuring gender bias in word embeddings across domains and discovering new gender bias word categories. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 25–32, 2019.

[19] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings 13*, pages 82–102. Springer, 2013.

[20] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *University of Science and Technology of China and BAAI*, 2024.

[21] Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5747–5758, 2023.

[22] Myra Cheng, Esin Durmus, and Dan Jurafsky. Marked personas: Using natural language prompts to measure stereotypes in language models. In *ACL*, pages 1504–1532, 2023.

[23] Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*, 2023.

[24] Nicholas A Christakis and James H Fowler. *The spread of obesity in a large social network over 32 years*. New England J. of Medicine, 2007.

[25] Poomrapee Chuthamsatid, Shera Potka, and Alex Thomo. Word embedding bias in large language models. In *I-SPAN 2025*. Springer, 2025.

[26] Federico Cinus, Marco Minici, Corrado Monti, and Francesco Bonchi. The effect of people recommenders on echo chambers and polarization. In *AAAI Conference on Web and Social Media*, pages 90–101, 2022.

[27] Google Cloud. Text embeddings api — generative ai on vertex ai. `https://cloud.google.com/vertex-ai/generative-ai/docs/model-reference/text-embeddings-api`, 2023.

[28] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. Privacy-preserving neural representations of text. *arXiv preprint arXiv:1808.09408*, 2018.

[29] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Academic Press, 2013.

[30] Cohere. Embed api reference. https://docs.cohere.com/reference/embed, 2023.

[31] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[32] Fernando Diaz, Michael Gamon, Jake Hofman, Emre Kıcıman, and David Rothschild. Online and social media data as a flawed continuous panel survey. In *PloS one*, volume 13, page e0190804, 2018.

[33] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[34] Christophe Dupuy, Radhika Arava, Rahul Gupta, and Anna Rumshisky. An efficient dp-sgd mechanism for large scale nlu models. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4118–4122. IEEE, 2022.

[35] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[36] Times Higher Education. World university rankings 2024. https://www.timeshighereducation.com/world-university-rankings/2024/world-ranking, 2024.

[37] Michael D Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. Fairness and discrimination in information access systems. *arXiv preprint arXiv:2105.05779*, 2021.

[38] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. *arXiv preprint arXiv:1808.06640*, 2018.

[39] Fatemeh Esfahani, Venkatesh Srinivasan, Alex Thomo, and Kui Wu. Nucleus decomposition in probabilistic graphs: Hardness and algorithms. In *ICDE*, pages 218–231. IEEE, 2022.

[40] Lisette Espín-Noboa, Claudia Wagner, Markus Strohmaier, and Fariba Karimi. Inequality and inequity in network-based ranking and recommendation algorithms. *Scientific reports*, 12(1):2012, 2022.

[41] Francesco Fabbri, Francesco Bonchi, Ludovico Boratto, and Carlos Castillo. The effect of homophily on disparate visibility of minorities in people recommender systems. In *AAAI Conference on Web and Social Media*, pages 165–175, 2020.

[42] Francesco Fabbri, Maria Luisa Croci, Francesco Bonchi, and Carlos Castillo. Exposure inequality in people recommender systems: the long-term effects. In *AAAI Conference on Web and Social Media*, pages 194–204, 2022.

[43] Antonio Ferrara, Lisette Espín-Noboa, Fariba Karimi, and Claudia Wagner. Link recommendations: Their impact on network structure and minorities. In *Proceedings of the 14th ACM Web Science Conference 2022*, pages 228–238, 2022.

[44] Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th international conference on web search and data mining*, pages 178–186, 2020.

[45] Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. Leveraging hierarchical representations for preserving privacy and utility in text. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 210–219. IEEE, 2019.

[46] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[47] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *WWW*, 2018.

[48] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864. ACM, 2016.

[49] Wei Guo and Aylin Caliskan. Detecting emergent intersectional biases: Contextualized word embeddings contain a distribution of human-like biases. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 122–133, 2021.

[50] Yang Guo, Fatemeh Esfahani, Xiaojian Shao, Venkatesh Srinivasan, Alex Thomo, Li Xing, and Xuekui Zhang. Integrative covid-19 biological network inference with probabilistic core decomposition. *Briefings in Bioinformatics*, 23(1):bbab455, 2022.

[51] Ivan Habernal. When differential privacy meets nlp: The devil is in the detail. In *ACL Anthology*, 2021.

[52] Joseph Howie, Venkatesh Srinivasan, and Alex Thomo. Scaling up structural clustering to large probabilistic graphs using lyapunov central limit theorem. *Proceedings of the VLDB Endowment*, 16(11):3165–3177, 2023.

[53] Timour Igamberdiev and Ivan Habernal. Dp-bart for privatized text rewriting under local differential privacy. In *ACL Anthology*, 2023.

[54] Ahmad Khajehnejad, Moein Khajehnejad, Mahmoudreza Babaei, Krishna P Gummadi, Adrian Weller, and Baharan Mirzasoleiman. Crosswalk: Fairness-enhanced node representation learning. In *AAAI*, 2022.

[55] Nikolay Korovaiko and Alex Thomo. Trust prediction from user-item ratings. *Social Network Analysis and Mining*, 3:749–759, 2013.

[56] Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. *arXiv preprint arXiv:1805.06093*, 2018.

[57] Jan Malte Lichtenberg, Alexander Buchholz, and Pola Schwöbel. Large language models as recommender systems: A study of popularity bias. *arXiv preprint arXiv:2406.01285*, 2024.

[58] Lingjuan Lyu, Xuanli He, and Yitong Li. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. *arXiv preprint arXiv:2010.01285*, 2020.

[59] Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. Towards differentially private text representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1813–1816, 2020.

[60] Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. The limits of word level differential privacy. In *ACL Anthology*, 2022.

[61] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[62] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[63] Microsoft. Presidio: Data protection and de-identification sdk, 2023. Accessed: October 15, 2023.

[64] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[65] Fatemehsadat Mireshghallah, Huseyin A Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. Privacy regularization: Joint privacy-utility optimization in language models. *arXiv preprint arXiv:2103.07567*, 2021.

[66] Giovanna Miritello, Rubén Lara, Manuel Cebrian, and Esteban Moro. Limited communication capacity unveils strategies for human interaction. *Scientific reports*, 3(1):1950, 2013.

[67] Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403, 2008.

[68] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the Nat. Academy of Sciences*, 103(23):8577–8582, 2006.

[69] OpenAI. Embeddings guide. `https://platform.openai.com/docs/guides/embeddings`, 2023.

[70] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[71] Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101, 2022.

[72] Shera Potka, Isla Li, Jason Kepler, and Alex Thomo. Enhancing structural minority visibility in link recommendations. In *MEDES 2024*. Springer, 2024.

[73] Shera Potka and Alex Thomo. Community structure and coherence in digital humanities works. In *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–8. IEEE, 2023.

[74] Tahleen A. Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. In *IJCAI*, 2019.

[75] L. Rainie and B. Wellman. *Networked: The new social operating system*. MIT Press, 2012.

[76] Akrati Saxena, George Fletcher, and Mykola Pechenizkiy. Nodesim: node similarity based network embedding for diverse link prediction. *EPJ Data Science*, 11(1):24, 2022.

[77] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *KDD*, pages 2219–2228, 2018.

[78] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Sst-2.

[79] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390, 2020.

[80] Ana-Andreea Stoica, Christopher Riederer, and Augustin Chaintreau. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. In *WWW*, pages 923–932, 2018.

[81] Jessica Su, Aneesh Sharma, and Sharad Goel. The effect of recommendations on network structure. In *WWW*, pages 1157–1167, 2016.

[82] Tan Thongtan and Tanasanee Phienthrakul. Sentiment classification using document embeddings trained with cosine similarity. In Fernando Alva-Manchego, Eunsol Choi, and Daniel Khashabi, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 407–414, Florence, Italy, July 2019. Association for Computational Linguistics.

[83] Autumn Toney-Wails and Aylin Caliskan. Valnorm quantifies semantics to reveal consistent valence biases across languages and over centuries. *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[84] Meng Tong, Kejiang Chen, Yuang Qi, Jie Zhang, Weiming Zhang, and Nenghai Yu. Privinfer: Privacy-preserving inference for black-box large language model. *arXiv preprint arXiv:2310.12214*, 2023.

[85] Vincent A Traag, Ludo Waltman, and Nees Jan van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

[86] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. Fairness-aware pagerank. In *WWW*, pages 3815–3826, 2021.

[87] Carl Vogel. Law matters, syntax matters and semantics matters. *Formal Linguistics and Law*, 212:25, 2009.

[88] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv*, 2212.03533, 2022.

[89] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. *Advances in neural information processing systems*, 30, 2017.

[90] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*, 2024.

[91] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. Differential privacy for text analytics via natural text sanitization. pages 3853–3866, 2021.

[92] Guixian Zhang, Guan Yuan, Debo Cheng, Lin Liu, Jiuyong Li, and Shichao Zhang. Mitigating propensity bias of large language models for recommender systems. *arXiv preprint arXiv:2409.20052*, 2024.

[93] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, 2017.