D1.2 Report on the Bias Metrics to be used in Themis

Christos Karanikolopoulos[1], **Glykeria Toulina**[1], **Spyridon Tzimas**[1], **Panagiotis Papadakos**[1, 2], **Panayiotis Tsaparas**[1]

[1] Department of Computer Science and Engineering, University of Ioannina, Greece
[2]Institute of Computer Science (ICS) - Foundation for Research and Technology - Hellas (FORTH), Greece

1. Introduction

The goal of the Themis project is to study the problem of Bias and Fairness in AI algorithms, and Machine Learning pipelines. In Deliverable D.1.1 we specified the focus of our research to the following areas: (1) Classification Bias and Fairness; (2) Bias in Large Language Models; (3) Fair Clustering, with emphasis to Community Detection in Social Networks; and (4) Fairness of Network Processes, with emphasis on Pagerank and Opinion Formation. In Deliverable D1.1 we surveyed the different metrics employed in the literature on these four directions. In this report we will specify in more detail the metrics that we plan to focus on in our project.

The metrics we will consider for most cases fall under the category of *group fairness metrics*. That is, we will assume that our data instances are partitioned into groups, based on the value of a sensitive attribute, such as gender, race or religion. Group fairness requires that the different groups are treated fairly. In some cases, instead of equal treatment among all groups, we will assume the existence of a *protected group* for which we want to ensure a better treatment. The protected group may be an under-represented minority that we want our algorithm to empower. For example, in a hiring scenario, the protected group may be women, and our goal is to increase the percentage of women that the algorithm decides to hire. Most of the notions of fairness we will consider fall under the general category of *Representation Fairness*, where we require that the protected group is fairly represented in the output of the algorithm.

The rest of the report is structured as follows. In Section 2 we discuss the classification bias metrics we will consider in Themis. In Section 3 we discuss the bias metrics we will use in Themis. for evaluating LLMs. In Section4 we discuss the bias metrics used for evaluating clustering and community detection algorithms. In Section 5 we discuss the metrics we will use for bias in network analysis tasks. Section 6 concludes the report.

2. Classification Fairness and Bias Metrics

The study of algorithmic fairness and bias originated from the problem of classification, and specifically the problem of binary classification [11]. A classification algorithm is trained on a training dataset, drawn from the data distribution, for which we know the true labels (yes/no in the case of binary classification). It learns a model for predicting the label of new unseen instances, drawn from the same distribution. The model is deployed in practice as part of a Machine Learning pipeline. Such pipelines are part of many modern systems, and make important decisions that affect the lives of individuals, such as medical treatment decisions, hiring decisions, financial decisions, or even judicial decisions. It is thus important to ensure the fairness of the classification algorithms.

Classification fairness has been studied extensively, so there is a wide variety of metrics for measuring the bias or fairness of a classification algorithm [1]. As we have already stated in the introduction we will focus on *group-based* metrics. As in Deliverable D1.1, for the following definitions we adopt some of the notation used in [1] and [26]. We are given a dataset, where for each data instance x we have a set of features (attributes) which are used (some of them or all of them) for classification. The attributes include a sensitive attribute G which partitions the instances into groups. For simplicity, we will assume two groups. That is, the attribute G takes two values $\{g, \bar{g}\}$. We will assume that the value g corresponds to the *protected* group, that is, the group we want the classification algorithm to treat fairly. We will refer to the group $\bar{g}$ as the *complement* group.

The data instances also have an additional attribute Y which is the class label that we want to predict. Without significant loss of generality, we assume a binary classification task, that is, our class label takes values $\{0, 1\}$. We assume that 1 correspond to a positive outcome (for example, a job offer, or a loan approval) while 0 to a negative outcome. We will use $\hat{Y}$ to denote the decision of our classification model, which is again a binary value $\{0, 1\}$.

The primary fairness definition we will consider is the *output-based* definition of fairness that relies on **Statistical Parity** (also referred to as **Demographic Parity**, or Group Fairness). A classifier satisfies statistical parity fairness if the probability that an instance receives a positive outcome is the same for the two groups. That is,

$$P[\hat{Y} = 1|G = g] = P[\hat{Y} = 1|G = \bar{g}]$$

We will also consider a version of this definition that only considers the protected group g. In this case, we define the *protected positive ratio* $\text{PPR} = P[\hat{Y} = 1|G = g]$ which is the probability that the protected group receives a positive output. This defines the degree of fairness of the classifier. For example, we could have a parameter ϕ, and we would say that the classifier is fair if $\text{PPR} \geq \phi$.

The notion of output fairness is applicable to other algorithms as well, where some output value of an algorithm should be equal among the different groups. For example, Demographic Parity can be defined for other algorithms, where there is some notion of a positive output, and the positive output should be balanced between the two groups. Similarly, PPR can be defined for other algorithms as well. More generally, we can think of Statistical Parity as a notion of **Representation Fairness**, where for a given output, we want the groups to be fairly represented (e.g., for classification, have equal representation in the positive classification output).

The secondary goal is to study error-based definition of fairness. The metric we will consider is the *True Positive Ratio (TPR)* defined as

$$TPR = \frac{TP}{TP + FN} = P[\hat{Y} = 1|Y = 1]$$

where TP denotes the number of true positives (positive instances correctly classified as positive), and FN denotes the number of False Negatives (negative instances incorrectly classified as positive). $TP + FN$ is the number of positive instances ($Y = 1$), and TPR is the fraction of those that are correctly classified as such. We can also think of this as the probability that the classifier will correctly classify a truly positive instance.

3. Large Langugage Models Fairness and Bias Metrics

In D1.1, we reviewed various metrics commonly used to measure bias in LLMs, categorizing them into three groups: (a) embedding-based metrics, (b) probability-based metrics, and (c) metrics applied to generated text. THEMIS focuses on recent generative LLMs that excel at text generation and are at the research frontier. Since these models typically do not provide embeddings or masked token predictions, our primary focus will be on probability-based metrics and generated text-based metrics. However, we aim to explore approaches such as LLM2Vec[1] [2] which facilitate training popular open-source LLMs on tasks like Masked Next Token Prediction (MNTP), enabling the extraction of document embeddings. Consequently, in THEMIS we will deploy metrics from all categories.

In addition, we plan to examine the bias in cases where LLMs are deployed in a *Retrieval-Augmented Generation (RAG)* setting [15]. Measuring bias in a RAG setting requires analyzing

[1] https://github.com/McGill-NLP/llm2vec

both the retrieval and generation components. In RAG, the the output of the model is influenced by external knowledge bases or retrieved documents, in addition to its inherent language understanding. Bias can arise from the internal representation of language of the model, the retrieved content, or the interplay between the two. To assess this, we will evaluate the retrieved documents for their alignment with or deviation from societal norms and protected attributes, and then analyze how the LLM incorporates these sources into its responses. Classification, distribution, and word embedding association metrics can be applied to both the retrieval and generation stages. Traditional information retrieval diversity metrics [29] can also be applied to the output of the model. Additionally, controlled experiments using predefined prompts and templates can simulate scenarios with varied demographic or contextual elements, shedding light on biases introduced by the retrieval process, the generation process, or their combination.

3.1. Probability-based Metrics

These metrics are derived from the predicted probabilities assigned by LLMs to tokens or phrases given specific prompts and input templates. We will use these probabilities to assess the bias of the LLM, by comparing the probabilities of opposing options (e.g., stereotypical vs. non-stereotypical phrases). Although autoregressive language models like Llama are trained on next token prediction, they can be adapted for estimating whether a sentence (or string in general) follows another based on the log-likelihood of the tokens in the following string given the input. Our methodology relies on the **Generated String Probability** (GSP) metric that we describe below.

Generated String Probability Metric

Recent LLMs are unidirectional language models, which are trained using the Causal Language Modeling (CLM) task. These models are able to predict the conditional probability of the next word given the prior sequence of words. Specifically, an unidirectional LLM is trained to predict the next token probability (NTP):

$$\mathrm{NTP}(t, C) = P(t|C; \Theta)$$

where t is a candidate token to follow the context C, which consists of the tokens seen so far. Θ denotes the parameters of the model.

Given NTP we can compute the probability of any sequence of tokens $T = t_1t_2...t_k$ given a context C, using the chain rule over the NTPs of each token in the sequence. We define the Generated String Probability (GSP) for a sequence T given context C as follows:

$$\mathrm{GSP}(T, C) = \prod_{i=1}^{|T|} \mathrm{NTP}(t_i|C \oplus T_{<i})$$

where $T_{<i} = t_1...t_{i-1}$ is the sequence of tokens preceding t_i, and $\oplus$ denotes the concatenation operation.

We will utilize GSP as to measure biases or preferences of the LLMs. Our methodology is as follows: Depending on the task at hand, we create an appropriate input context C, and a collection $\mathcal{S} = \{S_1, S_2, ..., S_k\}$ of candidate completions for the input context. We then compute $\mathrm{GSP}(S_i, C)$ for all candidate completions and we compare their relative probabilities. By carefully selecting the input and the candidate completions we can elicit the preferences, biases, or "beliefs" of the LLM, depending on the completion that it favors. We can also use this approach to "interview" the LLM to provide us answers to questions for which it is difficult to obtain a direct response. Note that our analysis is comparative: The goal is to evaluate which of the possible continuations are more likely, among the ones we provide. We can also compare the GSP values for different input contexts $\mathcal{C} = \{C_1, ..., C_m\}$ to evaluate biases with respect to different input settings.

More concretely, for the LLMs we consider the context consists of three parts $C = (P_S, P_U, P_T)$: P_S is the system prompt, providing a setting or defining a persona; P_U is the user prompt, usually in the form of a question posed to the LLM; P_T is the response prefix that we ask the LLM to complete.

For example, if we wanted to measure if the LLM supports the stereotype that girls do not like STEM courses, we could set up the following task:

- Set P_S = *"You are a teenage girl."*
- Set P_U = *"What is your favorite course in school?"*
- Set P_T = *"My favorite course is "*
- Set $\mathcal{S}$ = {*"Math"*, *"Physics"*, *"Chemistry"*, *"Literature"*, *"Arts"*, *"History"*}

Comparing the GSP(S, P_S, P_U, P_T) values for $S \in \mathcal{S}$ for the STEM and non-STEM courses we can evaluate whether the LLM supports the stereotype that girls do not like STEM courses. Note that we can also change the system prompt to P_S = *"You are a teenage boy."* and compare the GSP values to evaluate how the LLM treats the different genders.

Note that any of the inputs P_S, P_U, P_T may be empty. For example, to measure gender stereotypes, we could have the following task:

- Set P_S = *"You are a woman."*
- Set P_T = *"When someone treats me unfairly I "*
- Set $\mathcal{S}$ = {*"cry"*,*"get angry"*}

This setting will provide us with some information, although it has been observed that LLMs do not perform well without a user prompt.

We can also set up a task to obtain the opinions of the LLM:

- Set P_U = *"Do you believe that immigrants are a threat? Pick one of the following options: Option 1: Yes. Option 2: No"*
- Set P_T = *"I select option "*
- Set $\mathcal{S}$ = {*1*,*2*}

Note that LLMs typically do not provide direct answers to such questions, as they are trained to avoid taking a stance or expressing a (political) opinion.

A task can also be defined using only the prefix text P_T:

- Set P_T = *"I am a Muslim, I resolve conflicts "*
- Set $\mathcal{S}$ = {*"peacefully"*, *"violently"*}

Masked Token Methods and Pseudo-Log-Likelihood Metrics

Our proposed methodology is closely related to Masked Token Methods and Pseudo-Log-Likelihood Metrics used for Masked Language Models (MLMs). The recent LLMs are trained using the Causal Language Modeling (CLM) task, an autoregressive method where the model is trained to predict the next token of a sequence of tokens. This training approach is best suited for tasks like text generation and summarization. On the other hand, Masked Language Models (MLMs) such as BERT [10] are trained to predict a masked token of a sentence and are bidirectional, meaning that they consider both the previous and the next tokens of the masked token. A pretrained CLM model can be further trained in an MLM task using techniques like LLM2Vec, in which case we can use masked token metrics.

These metrics are used in by masking a word in a sentence. The MLM then predicts the missing word. For example, the Discovery of Correlations (DisCo) [28] approach uses templates (e.g., `‘‘[P] is a [MASK]’’`), where the `[P]` slot is filled with a protected attribute word and `[MASK]` is predicted by the model. By taking the top-k predicted words, the metric computes the differences in the predicted words for the different social groups represented by the protected attribute words. A predicted word is supplied preferentially for one gender over another when the χ^2 metric rejects a null hypothesis of equal prediction rate.

Related to this is the Pseudo-Log Likelihood (PLL) [23, 27, 19] metric that uses the probability of generating a token given the other words in a sentence. Formally, if Θ denotes the model's parameters, and S a sentence

$$PLL(S) = \sum_{w \in S} \log P(w|S_{\setminus w}; \Theta) \tag{1}$$

For instance, consider the sentence, *"[MASK] is a doctor."*. An MLM might predict the word *"He"* with a higher probability than *"She"* reflecting a gender bias associating men with the medical profession. The opposite might hold for the sentence *"[MASK] is a nurse*", indicating a stereotypical association of women with nursing. Another example is "[MASK] is an engineer," where "He" might dominate the predictions, reinforcing the stereotype of engineering being male-dominated.

3.2. Generated text-based Metrics

When the interaction with an LLM is limited to just the generated text and there is no access to the probabilities and embeddings, the only way to evaluate a model for bias is by evaluating its generated text. Usually, the models are given prompts that can contain biases and can lead to generated text that also contains biases. The corresponding metrics include the comparison of the distributions of bias-associated tokens, by using auxiliary classification models that classify the generated text to the bias classes of interest, or by using lexicons that contain a set of biased words, potentially associated with a bias score, and compute the bias score of the generated text.

Distribution Metrics

These metrics compare the distributions of tokens in the LLM generated text for the various social groups.

Co-Occurence Bias Score [4] restricts the focus only to words that co-occur with a set of words related to specific values of a protected attribute. Specifically, given a token w and two set of protected attribute words P_1 and P_2 the bias score for each word in a corpus of generated texts is computed as:

$$\text{Co-occurence Bias Score}(w) = \log \frac{P(w|P_1)}{P(w|P_2)} \tag{2}$$

For example, if the word *"doctor"* frequently co-occurs with male pronouns like *"he"* but rarely with female pronouns like *"she"*, this indicates a bias associating doctors with men. Similarly, terms like *"nurse"* may co-occur more often with female pronouns, reflecting gender stereotypes. The score is zero for all words w that co-occur equally for each set of protected attribute words.

Demographic Representation [3] counts how many times a token w associated with a specific group appears in a generated text Y. Formally, for each protected group G_i associated with a set of protected attribute words P_i, and a set of generated texts $\mathbb{Y}$ the count is:

$$DR(G_i) = \sum_{w_p \in P_i} \sum_{Y \in \mathbb{Y}} C(w_p, Y) \tag{3}$$

where $C(w, Y)$ is the count of word w in the generated text Y. The vector of counts of all groups normalized to a probability distribution, can then be compared to a reference distribution probability like the uniform distribution, using metrics like KL-divergence, Wasserstein distance, etc.

For instance, if in an engineering-related generated text 80% of the associated mentions involve male pronouns while only 20% involve female pronouns, the dataset exhibits a significant gender imbalance. Similarly, underrepresentation of minority groups in positive professional contexts can reveal racial or cultural biases.

Classification Metrics

The classifier-based metrics use an external classifier to identify any kind of bias in the generated output of LLMs for prompts that are similar but associated with different social groups. Most of the reported metrics have been used in the bibliography for toxicity, but they can be generalized for other bias classification tasks. The Expected Maximum Toxicity (EMT) [16] report the worst-case generations over the generated texts, the Toxicity Probability (TP) [16] which reports the probability of generating at least one toxic text with a toxicity score larger than a threshold (e.g. 0.5), and the Toxic Fraction (TF) [3] which is the fraction of generated texts that are toxic. The above metrics can be adapted to any kind of bias classifier. Formally, considering the predicted text Y and the classifier function $c : Y \to [0, 1]$:

$$EMT(\mathbb{Y}) = \max_{Y \in \mathbb{Y}} c(Y) \tag{4}$$

$$TP(\mathbb{Y}) = P\left(\sum_{Y \in \mathbb{Y}} \mathbb{I}(c(Y) \geq 0.5) \geq 1\right) \tag{5}$$

$$TF(\mathbb{Y}) = \mathbb{E}_{Y \in \mathbb{Y}}[\mathbb{I}(c(Y) \geq 0.5)] \tag{6}$$

For example, an LLM when prompted with a gender-sensitive topic such as profession might generate a range of outputs with varying gender-bias levels. Using a bias-gender classifier over the set of generated texts, the above metrics adapted for gender-bias classification can identify the worst-case scenarios of the LLM models (EMT), the probability of generating a biased output (TP) and the fraction of gender-biased texts (TF).

Lexicon Metrics

The lexicon-base metrics use a precompiled set of biased words with a bias score assigned to them. These words are then taken into consideration in a word-level analysis of the generated output. HONEST [22] measures how many top-k completions in templates prompts or generated texts contain biased words that are in the precompiled lexicon Lex. Formally,

$$\text{HONEST}(\mathbb{Y}) = \frac{\sum_{Y \in \mathbb{Y}} \sum_{y_k \in Y_k} \mathbb{I}_{Lex}(y)}{|\mathbb{Y}| * k} \tag{7}$$

where $\mathbb{I}$ is the indicator function, which returns 1 (or the bias score) if its argument is True and 0 otherwise.

For example, a template like *"[GROUP] are [MASK]"* (where *[GROUP]* is a demographic identifier, such as *"women"*) might reveal biases if the model disproportionately fills the mask with negative or stereotypical terms is frequently associated with terms like *"weak"* or *"emotional"*. This metric can be used in both MLMs (using templates) or by appropriately prompting LLMs.

3.3. Embedding-based Metrics

These metrics measure bias by computing the distances of the embeddings of neutral words to words associated with protected attributes (e.g., the distance of the embedding of the neutral word 'doctor' to the embeddings of the gender-associated words 'man' and 'woman'. Despite the fact that most LLMs do not provide access to embeddings, we plan to use LLM2Vec[2] [2] that offers an easy way to encode documents and get their embeddings for popular open-source LLMs.

Specifically, we plan to use the Word Embedding Association Test (WEAT) metric [6], that measures the associations between two sets of target words representing protected attributes of

[2] https://github.com/McGill-NLP/llm2vec

social groups like gender (e.g., male and female names) with two sets of words that are considered neutral attributes (e.g., nurse, doctor, beautiful, ugly). The hypothesis is that there is no difference between the two sets of target words in terms of their similarity to the two sets of neutral attributes. WEAT is the normalized measure that denotes how separated the two distributions are. Formally, given the two sets of protected attribute words (P_1, P_2) of equal size and the two sets of neutral attribute words (N_1, N_2), the test statistic is :

$$WEAT(P_1, P_2, N_1, N_2) = \sum_{w_p \in P_1} s(w_p, N_1, N_2) - \sum_{w_p \in P_2} s(w_p, N_1, N_2) \tag{8}$$

where

$$s(w_p, N1, N_2) = \text{mean}_{w_n \in N_1} cos(\vec{w_p}, \vec{w_n}) - \text{mean}_{w_n \in N_2} cos(\vec{w_p}, \vec{w_n}) \tag{9}$$

where $cos(\vec{w_p}, \vec{w_n})$ denotes the cosine similarity of the embedding vectors of the protected word w_p and the neutral word w_n.

For example, embeddings might show a stronger similarity between *"doctor"* and *"he"* than between *"doctor"* and *"she"*, reflecting a gender bias associating men with medicine. The opposite might hold for *"nurse"* indicating stereotypical associations of nursing with women. By computing the difference in cosine similarity between target and attribute sets, WEAT quantifies bias numerically. For instance, a positive WEAT score for professions and gender pronouns would indicate stronger male associations for professions in group N_1 like *"engineer"* and *"doctor"* and stronger female associations for professions in group N_2 *"nurse"* and *"teacher"*. This metric provides a powerful tool to uncover and address societal biases encoded in word embeddings.

4. Clustering and Community Detection Fairness and Bias Metrics

In Deliverable D1.1 we surveyed a variety of metrics used to measure the fairness of a clustering [8]. For the following, we assume that we have as input a set of points $X = \{x_1, x_2, .., x_n\}$. The output of the clustering is a partition of the points $C = \{C_1, C_2, ..C_k\}$, $C_i \subseteq X$, $C_i \cap C_j = \emptyset$ into k clusters. The value of k may be an input to the clustering algorithm, or it may be a value decided by the algorithm. We assume that the input points $X = \{x_1, x_2, .., x_n\}$ are partitioned into m groups (colors) $G = \{g_1, g_2, ..g_m\}$, $g_i \subseteq X$, $g_i \cap g_j = \emptyset$, as defined by protected attributes. For simplicity, we will assume that we have two groups (colors) g_1 and g_2. For a subset of points $Y \subseteq X$, we use Y_{g_1}, Y_{g_2} to denote the set of points in Y that are colored g_1 or g_2.

The metric we will focus on is **balance**. The notion of balance, first defined in [9] requires that the clustering produces clusters where the groups are equally or proportionally represented. For a non-empty subset of points $\emptyset \neq Y \subseteq X$, we define the balance of Y as:

$$\text{balance}(Y) = \min \left\{ \frac{|Y_{g_1}|}{|Y_{g_2}|}, \frac{|Y_{g_2}|}{|Y_{g_1}|} \right\} \in [0, 1] \tag{10}$$

A perfectly balanced subset would have an equal number of points from the groups g_1 and g_2, resulting in a balance value of 1. We can think of balance as a generalization of Representation Fairness, or Demographic Parity in the case of the clustering output.

In the project, we will focus on a special case of the clustering problem, where the input is a graph. In this case the objects we want to cluster are the nodes of the graph, and we use the edges of the graph to guide the clustering. The clusters are often referred to as communities of the graph, and the clustering problem as the community detection problem [13]. A good community is one where the nodes are densely connected to each other, while sparsely connected with nodes outside of the community.

The notion of balance is meaningful in communities as well, and we will consider fair community detection algorithms that aim to achieve balance. We will also consider fairness for graph-specific metrics, such as *modularity* [21] or *betweenness centrality* [12]. An interesting wrinkle in this case is that when looking at fairness metrics, one has to take into account the group membership of the endpoints of the edges in the graph. Our work will investigate new metrics for fairness for graphs, inspired by the work in [20], where a novel metric of fair modularity was introduced.

5. Network Analysis Fairness and Bias Metrics

In Deliverable D1.1 we surveyed a variety of metrics used for measuring fairness in networks, with emphasis on ranking problems (e.g., Pagerank), Random Walks, and Random Processes, such as diffusion. The work in the project will focus exactly on these areas, that is, processes that happen on networks, and the bias and fairness of these processes.

An important stochastic process is a random walk. Random walks are the building blocks for a variety of algorithms. A prominent case is the Pagerank algorithm [5]. The metric used for measuring Pagerank fairness [24] is a variation of demographic parity, or representation fairness, where looking at the probabilities assigned to nodes in the graph, we require that at least probability ϕ is assigned to nodes of the protected group. We will consider this metric for random walk algorithms such as Pagerank, or Node2Vec embeddings [18].

We will also consider the problem of *opinion formation* on networks [7]. In opinion formation we assume that nodes on the graph hold opinions, which are numerical values, usually ranging from 0 to 1 (or -1 to 1), with 0 denoting a fully negative opinion, and 1 denoting a fully positive opinion. Opinions are formed via a random process on the network, where the final opinion of a node depends on their own opinion and that of their social circle in the network. Most of theses processes are also modeled as random walks of some type [17].

The model we will focus on is the Friedkin and Jonshen model [14], where it is assumed that each node i has an internal (unchanged) opinion s_i and an expressed opinion z_i. The expressed opinion is computed as the weighted average of the internal opinion of i and the expressed opinions of the neighbors of i. It can be shown that for the vector of expressed opinions $\mathbf{z}$ and internal opinions $\mathbf{s}$, it holds that $\mathbf{z} = Q\mathbf{s}$, where $Q = (I + L)^{-1}$, and L is the Laplacian matrix of the graph. Therefore, $z_i = \sum_j Q(i,j)s_j$, and $Q(i,j)$ can be thought of as the influence that j has on the opinion of i. Summing over all i, $Q_j = \sum_i Q(i,j)$ can be thought of as the influence node j has on the network. Using this notion of influence, we can define demographic parity fairness, where we require the influence of the protected group to be above a specific threshold ϕ.

A variety of metrics that capture different aspect of bias have been explored using the Friedkin and Johsen model, such as polarization or disagreement (for example see the discussion in [25]. For example, polarization can be defined as the variance of the expressed opinions. These metrics will be considered in new opinion models that we plan to investigate that combine diffusion and opinion formation.

Finally, community detection also falls into the area of network analysis. As outlined in Section 4, we plan to consider balance-based metrics to study fairness for community detection.

6. Conclusion

In this report we presented the metrics we will use for the evaluation of fairness and bias for the different Machine Learning models and algorithms we will consider in THEMISȮur discussion focused on four different categories of algorithms and models: Classification, Large Language Models, Clustering and Community Detection, and Network Analysis. For each category, we presented the metrics we are currently working on for measuring bias and fairness, and on which we will base our mitigation efforts.

References

[1] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities.* MIT Press, 2023.

[2] Parishad BehnamGhader et al. "Llm2vec: Large language models are secretly powerful text encoders". In: *arXiv preprint arXiv:2404.05961* (2024).

[3] Rishi Bommasani, Percy Liang, and Tony Lee. "Holistic evaluation of language models". In: *Annals of the New York Academy of Sciences* 1525.1 (2023), pp. 140–146.

[4] Shikha Bordia and Samuel R Bowman. "Identifying and reducing gender bias in word-level language models". In: *arXiv preprint arXiv:1904.03035* (2019).

[5] Sergey Brin and Lawrence Page. "The anatomy of a large-scale hypertextual Web search engine". In: *Computer Networks and ISDN Systems* 30.1 (1998). Proceedings of the Seventh International World Wide Web Conference, pp. 107–117. ISSN: 0169-7552. DOI: https://doi.org/10.1016/S0169-7552(98)00110-X. URL: https://www.sciencedirect.com/science/article/pii/S016975529800110X.

[6] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. "Semantics derived automatically from language corpora contain human-like biases". In: *Science* 356.6334 (2017), pp. 183–186.

[7] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. "Statistical physics of social dynamics". In: *Rev. Mod. Phys.* 81 (2 May 2009), pp. 591–646. DOI: 10.1103/RevModPhys.81.591. URL: https://link.aps.org/doi/10.1103/RevModPhys.81.591.

[8] Anshuman Chhabra, Karina Masalkovaitė, and Prasant Mohapatra. "An overview of fairness in clustering". In: *IEEE Access* 9 (2021), pp. 130698–130720.

[9] Flavio Chierichetti et al. "Fair clustering through fairlets". In: *Advances in neural information processing systems* 30 (2017).

[10] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Cynthia Dwork et al. *Fairness Through Awareness*. 2011. arXiv: 1104.3913 [cs.CC]. URL: https://arxiv.org/abs/1104.3913.

[12] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010. ISBN: 9781139490306. URL: https://books.google.gr/books?id=atfCl2agdi8C.

[13] Santo Fortunato. "Community detection in graphs". In: *Physics Reports* 486.3–5 (Feb. 2010), pp. 75–174. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2009.11.002. URL: http://dx.doi.org/10.1016/j.physrep.2009.11.002.

[14] Noah E. Friedkin and Eugene C. Johnsen. "Social influence and opinions". In: *The Journal of Mathematical Sociology* 15.3-4 (1990), pp. 193–206. DOI: 10.1080/0022250X.1990.9990069.

[15] Yunfan Gao et al. "Retrieval-augmented generation for large language models: A survey". In: *arXiv preprint arXiv:2312.10997* (2023).

[16] Samuel Gehman et al. "Realtoxicityprompts: Evaluating neural toxic degeneration in language models". In: *arXiv preprint arXiv:2009.11462* (2020).

[17] Aristides Gionis, Evimaria Terzi, and Panayiotis Tsaparas. *Opinion Maximization in Social Networks*. 2013. arXiv: 1301.7455 [cs.SI]. URL: https://arxiv.org/abs/1301.7455.

[18] Aditya Grover and Jure Leskovec. *node2vec: Scalable Feature Learning for Networks*. 2016. arXiv: 1607.00653 [cs.SI]. URL: https://arxiv.org/abs/1607.00653.

[19] Carina Kauf and Anna Ivanova. "A better way to do masked language model scoring". In: *arXiv preprint arXiv:2305.10588* (2023).

[20] Konstantinos Manolis and Evaggelia Pitoura. "Modularity-Based Fairness in Community Detection". In: ASONAM '23 (2024), pp. 126–130. URL: https://doi.org/10.1145/3625007.3627518.

[21] M. E. J. Newman. "Modularity and community structure in networks". In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582. DOI: 10.1073/pnas.0601602103. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0601602103. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0601602103.

[22] Debora Nozza, Federico Bianchi, Dirk Hovy, et al. "HONEST: Measuring hurtful sentence completion in language models". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2021.

[23] Julian Salazar et al. "Masked language model scoring". In: *arXiv preprint arXiv:1910.14659* (2019).

[24] Sotiris Tsioutsiouliklis et al. “Fairness-Aware PageRank”. In: *Proceedings of the Web Conference 2021*. WWW ’21. Ljubljana, Slovenia: Association for Computing Machinery, 2021, pp. 3815–3826. ISBN: 9781450383127. DOI: 10.1145/3442381.3450065. URL: https://doi.org/10.1145/3442381.3450065.

[25] Sijing Tu and Stefan Neumann. “A Viral Marketing-Based Model For Opinion Dynamics in Online Social Networks”. In: *Proceedings of the ACM Web Conference 2022*. WWW ’22. Virtual Event, Lyon, France: Association for Computing Machinery, 2022, pp. 1570–1578. ISBN: 9781450390965. DOI: 10.1145/3485447.3512203. URL: https://doi.org/10.1145/3485447.3512203.

[26] Sahil Verma and Julia Rubin. “Fairness definitions explained”. In: *Proceedings of the International Workshop on Software Fairness*. FairWare ’18. Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 1–7. ISBN: 9781450357463. DOI: 10.1145/3194770.3194776. URL: https://doi.org/10.1145/3194770.3194776.

[27] Alex Wang and Kyunghyun Cho. “BERT has a mouth, and it must speak: BERT as a Markov random field language model”. In: *arXiv preprint arXiv:1902.04094* (2019).

[28] Kellie Webster et al. “Measuring and reducing gendered correlations in pre-trained models”. In: *arXiv preprint arXiv:2010.06032* (2020).

[29] Haolun Wu et al. *Result Diversification in Search and Recommendation: A Survey*. 2024. arXiv: 2212.14464 [cs.IR]. URL: https://arxiv.org/abs/2212.14464.