Beyond ROUGE Scores in Algorithmic Summarization: Creating Fairness-Preserving Textual Summaries

Abhisek Dash
IIT Kharagpur, India

Anurag Shandilya
IIT Kharagpur, India

Arindam Biswas
IIT Kharagpur, India

Abhijnan Chakraborty
MPI SWS, Germany

Kripabandhu Ghosh
IIT Kanpur, India

Saptarshi Ghosh
IIT Kharagpur, India

ABSTRACT

As the amount of textual information grows rapidly, text summarization algorithms are increasingly being used to provide users a quick overview of the information content. Traditionally, summarization algorithms have been evaluated only based on how well they match human-written summaries (as measured by ROUGE scores). In this work, we propose to evaluate summarization algorithms from a completely new perspective. Considering that an extractive summarization algorithm selects a subset of the textual units in the input data for inclusion in the summary, we investigate whether this selection is *fair* or not. Specifically, if the data to be summarized come from (or cover) different socially salient groups (e.g., men or women, Caucasians or African-Americans), different political groups (Republicans or Democrats), or different news media sources, then we check whether the generated summaries *fairly represent* these different groups or sources. Our evaluation over several real-world datasets shows that existing summarization algorithms often represent the groups very differently compared to their distributions in the input data. More importantly, some groups are frequently *under-represented* in the generated summaries. To reduce such adverse impacts, we propose a novel fairness-preserving summarization algorithm 'FairSumm' which produces high-quality summaries while ensuring fairness. To our knowledge, this is the first attempt to produce fair summarization, and is likely to open up an interesting research direction.

KEYWORDS

Text summarization; Extractive summarization; Fair summarization; Fairness; Proportional representation

1 INTRODUCTION

Recently, there has been an explosion in the amount of information generated in the Web. To help Internet users deal with the information overload, text summarization algorithms are commonly used to get a quick overview of the textual information. Recognizing the business opportunities, many startups have mushroomed which offer content summarization services. For example, Agolo[1] provides a summarization platform to get most relevant information from both public and private documents. Aylien[2] or Resoomer[3] present relevant points and topics from a piece of text. Moreover, some news websites, e.g., Harvard Business Review[4], have started providing an 'executive summary' of the news articles, to help the readers lacking time to go through the full article. Multiple smartphone apps (e.g., News360, InShorts) have also been launched to provide short summaries of news stories.

A large number of summarization algorithms have been devised in the research community, which include algorithms for summarizing an individual large document, as well as summarizing a set of documents (e.g., a set of microblogs or news articles) – see [2] for a survey. Most of the summarization algorithms are *extractive* in nature, i.e., they form the summary by extracting some of the textual units in the input (e.g., individual sentences in a document, or individual microblogs in a set of microblogs) [15]. Additionally, some *abstractive* algorithms have also been devised, that attempt to generate natural language summaries [2]. In this work, we restrict our focus to extractive summarization and leave abstractive summarization as future work.

Extractive summarization algorithms essentially perform a selection of a (small) subset of the textual units in the input, for inclusion in the summary, based on some measure of the relative quality or importance of the textual units. Traditionally, these algorithms are judged on how closely the algorithmic summary matches gold standard summaries that are usually written by human annotators. To this end, measures such as ROUGE scores are used to evaluate the goodness of algorithmic summaries[20]. The underlying assumption behind this traditional evaluation criteria is that the data to be summarized is homogeneous, and the sole focus of summarization algorithms should be to identify *summary-worthy* information.

Information on the Web today is often an amalgamation of information of different classes, that come from multiple sources or groups, and often cover different perspectives. For example, on social media, different socially salient groups (e.g., men and women, Republicans and Democrats) discuss socio-political issues, and it is frequently observed that these social groups express very different opinions on the same topic or event [7]. Similarly, news media sources having different ideological leanings publish different articles on the same topic or event, covering different political parties, different gender issues, etc. Hence, while summarizing such heterogeneous data, one needs to check whether the generated summaries are properly representing the opinions of these different groups or sources. Therefore, in this paper, we propose to look at summarization algorithms from a completely new perspective. We propose to investigate whether the selection of the textual units in the summary is fair, i.e., *whether the generated summary fairly represents all the classes in the input data.*

Our investigation over four real-world datasets (two microblog datasets, and two DUC2006 datasets that are frequently used as

[1] https://www.agolo.com/splash
[2] https://aylien.com/text-api/summarization/
[3] https://resoomer.com/en/
[4] https://www.hbr.org

benchmarks for summarization) shows that many existing summarization algorithms do not fairly represent the input data in the generated summaries. Rather, some classes are systemically under-represented in the process. To reduce such unfairness, we develop a novel fairness-preserving summarization algorithm (named *FairSumm*) which selects highly relevant textual units in the summary while maintaining fairness in the process. Extensive evaluations show that FairSumm outperforms many existing summarization algorithms, not only in maintaining fairness, but also in providing better summary achieving high ROUGE scores.

In summary, we make the following contributions in this paper: (1) ours is one of the first attempts to consider the notion of fairness in summarization[5], (2) we show that existing summarization algorithms often do not fairly represent the input data, and (3) we propose 'FairSumm' algorithm which produces good quality and fair summary. Such a summary would not only benefit the end users of the summarization algorithms, but also many downstream applications that use the summaries generated by algorithms (e.g., Lloret *et al.* [22] proposed a summary-based opinion classification and rating inference mechanism). Such applications would also benefit from a fair summary, e.g., one which fairly represents the positive and negative opinions in the input data. We plan to make the implementation of FairSumm publicly available upon acceptance of the paper.

2 BACKGROUND AND MOTIVATION

In this section, we investigate whether existing summarization algorithms produce summaries that are fair. We experiment with two types of data containing textual units from multiple classes – (i) one where the different classes correspond to different values of a sensitive attribute, such as the political leaning of the textual units, or the gender of their authors, and (ii) the other where the different classes correspond to different sources having different viewpoints, such as different news media. For both types of datasets, we investigate whether the summaries produced by existing summarization algorithms represent the different classes fairly.

2.1 Summarizing data associated with a sensitive attribute

We apply several well-known extractive summarization algorithms on the following two microblog datasets where every textual unit is annotated with a sensitive attribute (gender or political leaning).

(1) **Claritin dataset**: This dataset contains tweets about the effects of the drug Claritin.[6] Each tweet is annotated with the gender of the user (male or female or unknown) who posted it. From this dataset, we ignored those tweets for which the gender of the user is unknown. The number of tweets in the different classes is reported in Table 1 (first row).

(2) **USelection dataset** [9]: This dataset contains tweets posted during the 2016 US Presidential elections. Each tweet is annotated as supporting or attacking one of the presidential candidates (Donald Trump and Hillary Clinton) or neutral or attacking both. For simplicity, we grouped the tweets into three classes: (i) *Pro-Republican*: tweets which support Trump and / or attack Clinton, (ii) *Pro-Democratic*: tweets which support Clinton and / or attack Trump, and (iii) *Neutral*: tweets which are neutral or attack both candidates. The number of tweets in the different classes is reported in Table 2 (first row).

Human-generated summaries: The traditional way of evaluating the 'goodness' of a summary is to match it with one or more human-generated summaries (gold standard), and then compute ROUGE scores [20]. To this end, we asked three human annotators to summarize the two datasets described above. Each annotator is well-versed with the use of social media like Twitter, is fluent in English, and none is an author of this paper. The annotators were asked to generate extractive summaries independently. We use these three human-generated summaries for the evaluation of various algorithmically-generated summaries, by computing ROUGE-1 and ROUGE-2 Recall and F_1 scores [20].

Summarization algorithms: We consider a set of well-known extractive summarization algorithms, some of which are unsupervised (the traditional methods for summarization) and some which are supervised neural models.

Unsupervised summarization algorithms: We consider six well-known summarization algorithms. These algorithms generally estimate an importance score for each textual unit (sentence / tweet) in the input, and k textual units having the highest importance scores are selected to generate a summary of length k.
(1) Cluster-rank [13] which clusters the textual units to form a cluster-graph, and uses graph algorithms (e.g., PageRank) to compute the importance of each unit.
(2) DSDR [18] which measures the relationship between the textual units using linear combinations and reconstructions, and generates the summary by minimizing the reconstruction error.
(3) LexRank [12], which computes the importance of textual units using eigenvector centrality on a graph representation based on similarity of the units, where edges are placed depending on the intra-unit cosine similarity.
(4) LSA [14], which constructs a terms-by-units matrix, and estimates the importance of the textual units based on Singular Value Decomposition on the matrix.
(5) LUHN [23], which derives a 'significance factor' for each textual unit based on occurrences and placements of frequent words within the unit.
(6) SumBasic [27], which uses frequency-based selection of textual units, and reweights word probabilities to minimize redundancy.

Supervised neural summarization algorithms: With the recently increasing popularity of neural network based models, the state of the art techniques for summarization have shifted to data-driven supervised algorithms [10]. We have considered two recently proposed extractive neural summarization models, proposed in [26]:
(7) SummaRuNNer-RNN, a Recurrent Neural Network (RNN) based sequence model that provides a binary label to each textual unit – a label of 1 implies that the unit can be part of the summary, while a label of 0 indicates otherwise. Each label has an associated confidence score. The summary is generated by picking textual units labeled 1 in decreasing order of their confidence score, until

[5] Fairness in summarization was briefly introduced in our prior work [32].
[6] The dataset is described in detail at *https://www.crowdflower.com/discovering-drug-side-effects-with-crowdsourcing/*.

Method	Nos. of tweets		ROUGE-1		ROUGE-2	
	Female	Male	Recall	F_1	Recall	F_1
Whole data	2,505 (62%)	1,532 (38%)				
ClusterRank	33	17	0.4369	0.4948	0.1614	0.1828
DSDR	31	19	0.3018	0.4251	0.1443	0.2033
LexRank	34	16*	0.2964	0.3926	0.1138	0.1599
LSA	35	15*	0.5153	0.5041	0.1506	0.1473
LUHN	34	16*	0.3802	0.4053	0.1280	0.1365
SumBasic	27*	23	0.3144	0.4341	0.1082	0.1494
SummaRNN	33	17	0.3423	0.3754	0.1257	0.1468
SummaCNN	30	20	0.3774	0.4087	0.1265	0.1460

Table 1: Results of summarizing the Claritin dataset: Number of tweets of the two classes, in the whole dataset and the summaries of length 50 tweets generated by different algorithms. Also given are the ROUGE-1 and ROUGE-2 Recall and F_1 scores of each summary. * indicates under-representation of a class according to the fairness notion of 'adverse impact' [4] (details in text).

Method	Nos. of tweets			ROUGE-1		ROUGE-2	
	Pro-Rep	Pro-Dem	Neutral	Recall	F_1	Recall	F_1
Whole data	1,309 (62%)	658 (31%)	153 (7%)				
ClusterRank	32	15	3	0.2472	0.3499	0.0611	0.0865
DSDR	28*	19	3*	0.2154	0.3313	0.0675	0.1039
LexRank	27*	20	3*	0.2525	0.3672	0.0788	0.1146
LSA	24*	20*	6	0.3107	0.4039	0.0832	0.1083
LUHN	34	13*	3*	0.2808	0.3754	0.0846	0.1131
SumBasic	27*	23	0*	0.1988	0.3111	0.0513	0.0803
SummaRNN	34	15	1*	0.3472	0.4361	0.1201	0.1601
SummaCNN	32	17	1*	0.3368	0.4227	0.1083	0.1446

Table 2: Results of summarizing the USelection dataset: Number of tweets of the three classes, in the whole dataset and the summaries of length 50 tweets generated by different algorithms. * indicates under-representation of a class according to 'adverse impact' [4].

the desired summary length is exceeded. The proposed model is built around a two-layer bi-directional Gated Recurrent Unit based Recurrent Neural Network (GRU-RNN).
(8) SummaRuNNer-CNN is a variant of the above model where the sentences are fed to a two layer Convolutional Neural Network (CNN) architecture before using GRU-RNN in the third layer.
For both the SummaRuNNer models, the authors have made the pretrained models available[7] which are trained on the CNN/Daily Mail news articles corpus[8]. We directly used the pretrained models for the summarization.

Results of summarization: We applied the above summarization algorithms on the two datasets, to obtain summaries of length 50 tweets each. Table 1 shows the results of summarizing the Claritin datasets, while Table 2 shows that for the USelection dataset. In both cases, shown are the numbers of tweets of the different classes in the whole dataset (first row), and in the summaries generated by the different summarization algorithms (subsequent rows), and the ROUGE-1 and ROUGE-2 recall and F_1 scores of the summaries.

Verifying if the summaries are fair: To check whether the generated summaries are fair, we apply the principle of *'adverse impact'* that is used by the U.S. Equal Employment Opportunity Commission to determine whether a company's hiring policy is biased against a demographic group [4]. According to this policy, a particular class c is **under-represented** in the selected set, if the fraction of selected items belonging to class c is less than 80% of the fraction of selected items of the class having the highest selection rate. Applying this rule, we find under-representation of particular classes of tweets in the summaries generated by many of the algorithms; these cases are marked with an asterisk (*) in Table 1 and Table 2.

We repeated the experiments for summaries of lengths other than 50 as well, such as for $100, 200, \ldots, 500$ (details omitted due to lack of space). We observed several cases where the same algorithm includes very different proportions of tweets of various classes, while generating summaries of different lengths. Hence, whether summarization is fair depends on several factors, including the particular algorithm used and the length of summary.

2.2 Summarizing data from multiple sources

For these experiments, we consider the standard summarization datasets provided by the Document Understanding Conference (DUC) 2006[9], which have been used to evaluate summarization algorithms in a large number of prior works [18]. The DUC06 datasets contains 50 'topics', and each topic contains 25 news articles relevant to the topic. The news articles are from three news media sources – The Associated Press (AP), The New York Times (NYTimes), and Xinhua News Agency (Xinhua). The DUC 2006 task was to generate a summary of 250 words out of all the articles in a topic. To evaluate the summaries, a set of gold standard summaries written by human assessors are also provided.

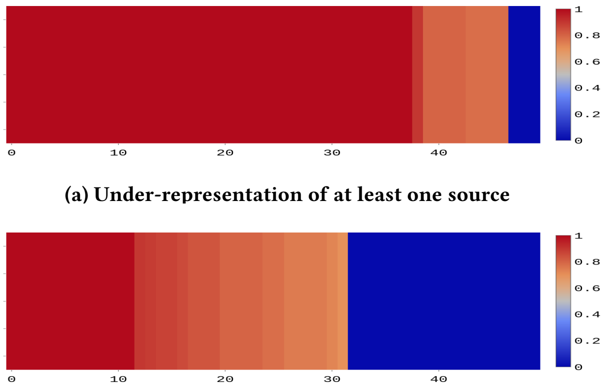
Several teams participated in the DUC 2006 task, and submitted summaries of length 250 words. The submitted summaries (called 'peers') are also provided along with the data. The submitted summaries contain both extractive and abstractive summaries. For the present work, we consider only the purely extractive summaries, i.e., those summaries in which every sentence is contained in one of the news articles for the corresponding topic. In total, there are 351 extractive summaries submitted across all the 50 topics.

Verifying if the summaries are fair: We checked whether the extractive summaries under-represent any of the three sources, according to the notion of 'adverse impact' described earlier. To this end, we consider individual sentences in the input articles / summaries as the textual units. Note that the sentences in each news article have already been separated in the datasets provided by DUC, making this a natural choice for textual units.

Out of the 351 extractive summaries, we found that 322 summaries (91%) under-represent at least one source, and 203 summaries (58%) under-represent two out of the three sources. Figure 1(a) shows the topic-wise distribution (across the 50 DUC06

[7] https://github.com/hpzhao/SummaRuNNer
[8] https://github.com/deepmind/rc-data

[9] https://duc.nist.gov/data.html



(a) Under-representation of at least one source

(b) Under-representation of two different sources

Figure 1: Unfairness in the DUC2006 submitted extractive summaries across the 50 topics – heat-maps show topic-wise distribution of under-representation. The cell corresponding to a particular topic is dark red if most summaries for this topic under-represent at least one source (in (a)) or under-represent two sources (in (b)), and of blue color if most summaries do not under-represent any source.

topics) of the fraction of summaries that under-represent at least one source. Similarly, Figure 1(b) shows the topic-wise distribution of summaries that under-represent two different sources.

These statistics show that a very large majority of the summaries submitted to DUC06 are not fair according to the notion of adverse impact. In fact, for as many as 12 out of the 50 topics, *all* the submitted extractive summaries under-represent two of the three sources. We choose the datasets for two such topics for further experiments later in the paper – D0621 (crime and law enforcement in China) and D0626 (bombing of US embassies in Africa). For these two topics, Table 3 and Table 4 show the number of sentences from the three sources in the input data, in some of the submitted extractive summaries (Peer XX) and in the extractive summaries generated by the various algorithms stated in Section 2.1. We also present the ROUGE-1 and ROUGE-2 Recall and F_1 scores, as computed using the gold standard summaries provided by DUC06. As before, under-represented sources are marked with an asterisk (*).

The experiments in this section establish that summaries generated by existing summarization algorithms are often not fair, and under-represent one or more classes of the textual units. Having identified the need for new algorithms for fair summarization, we proceed to define fairness notions for the problem of summarization which the algorithms should satisfy.

3 NOTIONS OF FAIR SUMMARIZATION

Next, we define some fairness notions in the context of summarization. As before, we assume that the textual units in the input data are categorized into different classes.

Equal Representation: According to the notion of *Statistical Parity* [24], a summarization algorithm will be fair if different classes in the input data are represented equally in the generated summary.

Proportional Representation: Often it may not be possible to satisfy equal representation of different classes in the summary, especially if the input data itself has very different proportions from the different classes. The notion of *Proportional Representation* requires that the representation of different classes in the summary should be proportional to their distribution in the input data.

Method	Nos. of tweets			ROUGE-1		ROUGE-2	
	APW	NYT	XIE	Recall	F_1	Recall	F_1
Whole data	167 (44%)	78 (21%)	131 (35%)				
Peer 2	1*	0*	6	0.3952	0.4008	0.0912	0.0905
Peer 24	1*	0	7*	0.4878	0.4691	0.1015	0.1017
Peer 27	5*	4	0*	0.3761	0.3542	0.0754	0.0766
ClusterRank	11	0*	0*	0.3718	0.3127	0.0702	0.0589
DSDR	6	0*	4	0.4509	0.4161	0.0875	0.0613
LexRank	5	3	2*	0.4356	0.4430	0.1078	0.0848
LSA	3*	3	4	0.4822	0.4021	0.0762	0.0677
LUHN	4*	0*	5	0.3974	0.3213	0.0935	0.0647
SumBasic	3*	3	3*	0.2847	0.3137	0.0437	0.0482
SummaRNN	10	0*	0*	0.4428	0.4272	0.0811	0.0637
SummaCNN	10	0*	0*	0.4328	0.4011	0.0712	0.0591

Table 3: Results of summarizing the DUC06 dataset, topic D0621: Number of sentences from the three sources, in the whole dataset and the summaries submitted by different participating teams. * indicates under-representation of a class.

Method	Nos. of tweets			ROUGE-1		ROUGE-2	
	APW	NYT	XIE	Recall	F_1	Recall	F_1
Whole data	338 (69%)	89 (18%)	65 (13%)				
Peer 2	4*	2*	2	0.4316	0.4097	0.1073	0.1018
Peer 24	7*	1*	2	0.4487	0.4251	0.1416	0.1341
Peer 27	5	1*	0*	0.4658	0.4648	0.1545	0.1541
Cluster Rank	8	0*	0*	0.3414	0.3355	0.0576	0.0566
DSDR	10	0*	0*	0.4411	0.3902	0.1324	0.1171
LexRank	5*	0*	4	0.4300	0.3648	0.1112	0.0942
LSA	6*	5	1*	0.4317	0.3445	0.1516	0.0981
LUHN	10	1*	0*	0.4005	0.3120	0.1516	0.0768
SumBasic	8*	1*	2	0.2870	0.3651	0.0616	0.0785
SummaRNN	10	0*	0*	0.4658	0.3772	0.1616	0.1202
SummaCNN	10	0*	0*	0.4529	0.3911	0.1472	0.1125

Table 4: Results of summarizing the DUC06 dataset, topic D0626: Number of sentences from the three sources, in the whole dataset and the summaries submitted by different participating teams. * indicates under-representation of a class.

No Adverse Impact: As defined earlier, the principle of *Adverse Impact* [4] is used to measure unfairness. We propose a notion of fairness, which would ensure *no adverse impact* in summarization. More specifically, this fairness notion requires that the fraction of textual units from any class, that is selected in the summary, *should not be* less than 80% of the fraction of selected units from the class having the highest selection rate (in the summary).

Symbols	Meanings
V	Set of textual units to be summarized
N	$\lvert V\rvert$, number of textual units to be summarized
t	Number of classes to which the textual units in V belong (e.g., $t = 2$ for Claritin dataset)
$Z_1, Z_2, \dots, Z_t$	The t classes of the textual units
S	Summary conforming to fairness notions ($S \subseteq V$)
k	Desired length of summary, $\lvert S\rvert$
$c_i, i = 1 \dots t$	Minimum number of textual units from class Z_i to be included in S (to satisfy fairness)
$\mathcal{F}$	Objective function (overall goodness measure of S)
$\mathcal{L}$	Coverage function (goodness measure of S)
$\mathcal{R}$	Diversity reward function (goodness measure of S)
$sim(i, j)$	Similarity score between two textual units $i, j \in V$
$\mathcal{M}$	A (partition) matriod
$\mathcal{I}$	Set of partitions of matriod $\mathcal{M}$
G	The optimized fair summary produced by Algorithm 1

Table 5: The main symbols used in Section 4.

4 FAIRSUMM: A FAIRNESS-PRESERVING SUMMARIZATION ALGORITHM

Our proposed fairness-preserving summarization algorithm treats summarization as an optimization problem of a submodular, monotone objective function, where the fairness requirements are applied as constraints. In this section, we first establish the theoretical premise of our algorithm, and then describe our algorithm. The symbols used in this section are given in Table 5.

4.1 Submodularity and Monotonicity

Definitions: Let $V = \{v_1, v_2, ..., v_n\}$ be the set of elements v_i each of which represents a textual unit (e.g., a tweet or a sentence) in the input collection to be summarized. We define a function $\mathcal{F} : 2^V \rightarrow \mathbb{R}$ (where $\mathbb{R}$ is the set of real numbers) that assigns a real value to a subset (say, S) of V. Our ultimate aim is to find S ($\subseteq V$) such that $|S| \leq k$, where $k \in \mathbb{R}$ is the desired length of the summary (that is specified as an input), and for which $\mathcal{F}$ is maximized (i.e. $S = argmax_{B \subseteq V} \mathcal{F}(B)$). So, from a set of textual units V, we look to find a summary S that maximizes an objective function $\mathcal{F}$.

Definition 1 (Discrete derivative) [19]: For $e \in V$ and a function $f : 2^V \rightarrow \mathbb{R}$, let $\Delta(e|S) = f(S \cup \{e\}) - f(S)$ be the *discrete derivative* of f at S with respect to e.

Definition 2 (Submodularity): a function f is *submodular* if for every $A \subseteq B \subseteq V$ and $e \in V \setminus B$ (i.e. $e \in V$ and $e \notin B$),

$$\Delta(e|A) \geq \Delta(e|B) \tag{1}$$

This property is popularly called the *property of diminishing returns.*

Definition 3 (Monotonicity): The function f is *monotone* (or *monotone nondecreasing* if for every $A \subseteq B \subseteq V$, $f(A) \leq f(B)$

Properties: We now discuss some of the important properties of monotone submodular functions which we will exploit in our problem formulation.

Property 1 (The class of submodular functions is closed under non-negative linear combinations): Let f_1, f_2, ..., f_n defined by $f_i : 2^V \rightarrow \mathbb{R}$ (i = 1, 2, ..., n) be submodular functions and λ_1, λ_2, ..., λ_n be non-negative real numbers. Then $\sum_{i=1}^{n} \lambda_i f_i$ is also submodular.
Proof: This is a well-known property of submodular functions (e.g., see [21]), hence the proof is omitted.

Property 2 (The class of monotone functions is closed under non-negative linear combinations): Let f_1, f_2, ..., f_n defined by $f_i : 2^V \rightarrow \mathbb{R}$ (i = 1, 2, ..., n) be monotone functions and λ_1, λ_2, ..., λ_n be non-negative real numbers. Then $\sum_{i=1}^{n} \lambda_i f_i$ is also monotone.
Proof: Let $A_1 \subseteq A_2 \subseteq V$. Then, since each f_i is a monotone, $f_i(A_1) \leq f_i(A_2)$. Let $F = \sum_{i=1}^{n} \lambda_i f_i$.
Case I: Let $\lambda_i = 0$, for all $i = 1, 2, \dots, n$. Then, $F = 0$ and is a constant (non-decreasing) function.
Case II: Let $\lambda_i > 0$, for all $i = 1, 2, \dots, n$. Then we can write

$$A_1 \subseteq A_2 \Rightarrow f_i(A_1) \leq f_i(A_2),\ \forall i \ \ \Rightarrow \sum_i^n f_i(A_1) \leq \sum_i^n f_i(A_2)$$

$$\Rightarrow \sum_i^n \lambda_i f_i(A_1) \leq \sum_i^n \lambda_i f_i(A_2) \ \ [\because\ \lambda_i > 0, \forall i]$$

$$\Rightarrow F(A_1) \leq F(A_2).$$

Case III: Let only some m ($m > 0$ and $m < n$) λ_is > 0. Let, without any loss of generality, such λ_is be λ_1, λ_2, ..., λ_m. Then we have

$$A_1 \subseteq A_2 \Rightarrow f_i(A_1) \leq f_i(A_2), \forall i \ \Rightarrow \sum_i^m f_i(A_1) \leq \sum_i^m f_i(A_2)$$

$$\Rightarrow \sum_i^m \lambda_i f_i(A_1) \leq \sum_i^m \lambda_i f_i(A_2)\ [\because \lambda_i > 0, \text{for } i = 1, 2, \dots, m)]$$

$$\Rightarrow \sum_i^n \lambda_i f_i(A_1) \leq \sum_i^n \lambda_i f_i(A_2)\ [\because \lambda_i = 0, \text{for } i = m + 1, m + 2, \dots, n)]$$

$$\Rightarrow F(A_1) \leq F(A_2)$$

This completes the proof.

Property 3 (The class of monotone submodular functions is closed under non-negative linear combinations): Let f_1, f_2, ..., f_n defined by $f_i : 2^V \rightarrow \mathbb{R}$ (i = 1, 2, ..., n) be monotone submodular functions and λ_1, λ_2, ..., λ_n be non-negative real numbers. Then $\sum_i \lambda_i f_i$ is also monotone submodular.
Proof: This follows trivially from Properties 1 and 2.

Property 4: Given functions F: $2^V \rightarrow \mathbb{R}$ and f: $\mathbb{R} \rightarrow \mathbb{R}$, the composition $F' = f \circ F : 2^V \rightarrow \mathbb{R}$ (i.e., $F'(S) = f(F(S))$) is nondecreasing sub-modular, if f is non-decreasing concave and F is nondecreasing submodular.
Proof: This is also a well-known property of submodular functions (e.g., see [21]), hence the proof is omitted.

4.2 Formulation of the objective function for summarization

We now look for an objective function for the task of extractive summarization. Following the formulation by Lin *et al.* [21], we use monotone submodular functions to construct the objective function. We consider the following two aspects of an extractive text summarization algorithm:

Coverage: Coverage refers to amount of information covered in the summary S, measured by a function, say, $\mathcal{L}$. The generic form of $\mathcal{L}$ can be

$$\mathcal{L}(S) = \sum_{i \in V, j \in S} sim_{i,j} \tag{2}$$

where $sim_{i,j}$ denotes the similarity between two textual units (tweets, sentences, etc.) $i \in V$ and $j \in V$. Thus, $\mathcal{L}(S)$ measures the overall similarity of the textual units included in the summary S with all the textual units in the input collection V.

Note that $\mathcal{L}$ is monotone submodular. $\mathcal{L}$ is monotonic since coverage increases by the addition of a new sentence in the summary. At the same time, $\mathcal{L}$ is submodular since the increase in $\mathcal{L}$ would be more when a sentence is added to a shorter summary, than when a sentence is added to a longer summary. There can be several forms of $\mathcal{L}$ depending on how $sim_{i,j}$ is measured, which we will discuss later in this paper.

Diversity reward: The purpose of this aspect is to avoid redundancy and reward diverse information in the summary. Let the associated function be denoted as $\mathcal{R}$. A generic formulation of $\mathcal{R}$ is

$$\mathcal{R}(S) = \sum_{j=1}^{K} \sqrt{\sum_{j \in P_i \cap S} r_j} \tag{3}$$

where $P_1, P_2, \ldots, P_K$ comprise a partition of V such that $\cup_i P_i = V$ and $P_i \cap P_j = \emptyset$ for all $i \neq j$; r_j is a suitable monotone submodular function that estimates the importance of adding the textual unit j to the summary. The partitioning $P_1, P_2, \ldots, P_K$ can be achieved by clustering the set V using any clustering algorithm (e.g., K-means), based on the similarity of items as measured by $sim(i, j)$.

$\mathcal{R}(S)$ rewards diversity since there is more benefit in selecting a textual unit from a partition (cluster) that does not yet have any of its elements included in the summary. As soon as any one element from a cluster P_i is included in the summary, the other elements in P_i will start having diminishing gains, due to the square root function.

The function r_j is a 'singleton reward function' since it estimates the reward of adding the singleton element $j \in V$ to the summary S. One possible way to define this function is:

$$r_j = \frac{1}{N} \sum_{i} sim(i, j) \tag{4}$$

which measures the average similarity of j to the other textual units in V.

Note that $\mathcal{R}(S)$ is monotone submodular by Property (4), since square root is a non-decreasing concave function. This formulation will remain monotone submodular if any other non-decreasing concave function is used instead of square root.

While constructing a summary, both coverage and diversity are important. Only maximizing coverage may lead to lack of diversity in the resulting summary and vice versa. So, we define our objective function for summarization as follows:

$$\mathcal{F} = \lambda_1 \mathcal{L} + \lambda_2 \mathcal{R} \tag{5}$$

where $\lambda_1, \lambda_1 \geq 0$ are the weights given to coverage and diversity respectively. Note that, by Property (3), $\mathcal{F}$ is monotone submodular.

Our proposed fairness-preserving summarization algorithm will maximize $\mathcal{F}$ in keeping with some fairness constraints. We now discuss this step.

4.3 Submodular Maximization using Partition Matroids

For fairness-preserving summarization, we essentially need to optimize the submodular function $\mathcal{F}$ while ensuring that the summary includes at least a certain number of textual units from each class present in the input data. This problem can be formulated using the concept of partition matroids, as described below.

Definitions: We start with some definitions that are necessary to formulate the constrained optimization problem.

Definition 4 (Matroid): A matroid is a pair $\mathcal{M} = (\mathcal{Z}, \mathcal{I})$, defined over a finite set (called the ground set) $\mathcal{Z}$ and a family of sets $\mathcal{I}$ (called the independent sets), that satisfies the following three properties:

(1) $\emptyset$ (empty set) $\in \mathcal{I}$.
(2) If $Y \in \mathcal{I}$ and $X \subseteq Y$, then $X \in \mathcal{I}$.
(3) If $X \in \mathcal{I}$, $Y \in \mathcal{I}$ and $|Y| > |X|$, then there exists $e \in Y \setminus X$ such that $X \cup \{e\} \in \mathcal{I}$.

Definition 5 (Partition Matroids): Partition matroids refer to a special type of matroids where the ground set $\mathcal{Z}$ is partitioned into disjoint subsets $\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_s$ for some s and
$\mathcal{I} = \{S \mid S \subseteq \mathcal{Z}$ and $|S \cap \mathcal{Z}_i| \leq c_i$, for all $i = 1, 2, ..., s\}$
for some given parameters $c_1, c_2, ..., c_s$. Thus, S is a subset of Z that contains at least c_i items from the partition $\mathcal{Z}_i$ (for all i), and $\mathcal{I}$ is the family of all such subsets.

Formulation of the constrained maximization problem: Consider that we have a set of control variables z_j (e.g., gender, political leaning), each of which takes t_j distinct values (e.g., male and female, democrat and republican). Each item in $\mathcal{Z}$ has a particular value for each z_j.

For each control variable z_j, we can partition $\mathcal{Z}$ into t_j disjoint subsets $\mathcal{Z}_{j1}, \mathcal{Z}_{j2}, ..., \mathcal{Z}_{jt_j}$, each corresponding to a particular value of this control variable. We now define a partition matriod $\mathcal{M}_j = (\mathcal{Z}, \mathcal{I}_j)$ such that

$$\mathcal{I}_j = \{S \mid S \subseteq \mathcal{Z} \text{ and } |S \cap Z_{ji}| \leq c_j, \text{for all } i = 1, 2, \ldots, t_j\}$$

for some given parameters $c_1, c_2, ..., c_{t_j}$.

Now, for a given submodular objective function f, a submodular optimization under the partition matriod constraints with P control variables can be designed as follows:

$$Maximize_{S \subseteq \mathcal{Z}} \quad f(S) \tag{6}$$

$$\text{subject to } S \in \textstyle\bigcap_{j=1}^{P} \mathcal{I}.$$

A prior work by Du *et al.* [11] has established that this submodular optimization problem under the matroid constraints can be solved efficiently with provable guarantees (see [11] for details).

4.4 Proposed summarization scheme

In the context of the summarization problem, the ground set is V (= $\mathcal{Z}$), the set of all textual units (sentences/tweets) which we look to summarize. The control variables (stated in Section 4.3) are analogous to the sensitive attributes with respect to which fairness is to be ensured. In this work, we consider only one sensitive attribute for a particular dataset (the gender of a user for the Claritin dataset, political leaning of a tweet for the USelection dataset, and the media source for the DUC06 dataset). Let the corresponding control variable be z, and let z take t distinct values (e.g., $t = 2$ for the Claritin dataset, and $t = 3$ for the USelection dataset). Note that, as described in Section 4.3, the formulation can be extended to multiple sensitive attributes (control variables) as well.

Each textual unit in V is associated with a class, i.e., a particular value of the control variable z (e.g., is posted either my a male or a female). Let $Z_1, Z_2, ..., Z_t$ ($Z_i \subseteq V$, for all i) be the disjoint subsets of the textual units from the t classes, each associated with a distinct value of z. We now define a partition matroid $\mathcal{M} = (V, \mathcal{I})$ in which V is partitioned into disjoint subsets $Z_1, Z_2, ..., Z_n$ and

$$\mathcal{I} = \{S \mid S \subseteq V \text{ and } |S \cap Z_i| \leq c_i, i = 1, 2, ..., t\}$$

for some given parameters $c_1, c_2, ..., c_t$. In other words, I will contain all the sets S containing at most c_i sentences from Z_i, $i = 1, 2, ..., t$.

Outside the purview of the matroid constraints, we maintain the restriction that c_i's are chosen such that
(1) $\sum_{i=1}^{t} c_i = k$ (the desired length of the summary S), and
(2) a desired fairness criterion is maintained in S. For instance, if equal representation of all classes in the summary is desired, then $c_i = \frac{k}{t}$ for all i.
We now express our fairness-constrained summarization problem as follows:

$$Maximize_{S \subseteq V} \mathcal{F}(S) \quad (7)$$
$$\text{subject to } S \in \mathcal{I}.$$

where the objective function $\mathcal{F}(S)$ is as stated in Equation 5.

The suitable algorithm to solve this constrained submodular optimization problem, proposed by Du et al. [11], is presented as Algorithm 1. The G produced by Algorithm 1 is the solution of Equation 7. This algorithm is an efficient alternative to the greedy solution which has a time complexity of $O(kN)$ where $N = |V|$ and $k = |S|$. On the other hand, it can be shown that the time complexity of Du et al. [11] is $O(min\{|G|N, \frac{N}{\delta} log \frac{N}{\delta}\})$, where δ is a factor (to be explained shortly). The reason for this efficiency is the fact that this algorithm does *not* perform exhaustive evaluation of all the possible submodular functions evolving in the intermediate steps of the algorithm. Instead, it keeps on decreasing the threshold w_t by a dividing factor $(1 + \delta)$, which skips the evaluation of many submodular functions and sets the threshold to zero when it is small enough. It selects elements z from the ground set (in our case, V) only at each threshold value to evaluate the marginal gain $(\mathcal{F}(G \cup \{z\})$ - $\mathcal{F}(G))$ without violating any constraints.

Note that, the theoretical guarantee of Algorithm 1 depends upon the number of partition matroids (P), i.e., the number of control variables, and the curvature (c_f) of $\mathcal{F}$ given by

$$c_f = max_{j \in V, \mathcal{F}(j)>0} \frac{\mathcal{F}(j) - \mathcal{F}(j|V \setminus \{j\})}{\mathcal{F}(j)}$$

The approximation ratio is $\frac{1}{P+c_f}$ (see [11] for details); in our setting, the number of partition matroids P is 1.

Algorithm 1 : Fairness-preserving summarization algorithm

```
Set $d = max_{z \in V} \mathcal{F}(\{z\})$.
Set $w_t = \frac{d}{(i+\delta)^t}$ for $t = 0, \ldots, l$ where $l = argmin_i\ [w_i \leq \frac{\delta d}{N}]$,
and $w_{l+1} = 0$.
Set $G = \emptyset$
for $t = 0, 1, \ldots, l, l + 1$ do
    for each $z \in$ and $G \cup \{z\} \in I$ do
        if $\mathcal{F}(G \cup \{z\})$ - $\mathcal{F}(G) \geq w_t$ then
            Set $G \leftarrow G \cup \{z\}$
        end if
    end for
end for
Output $G$
```

5 ALTERNATIVE MECHANISMS FOR FAIR SUMMARIZATION

In this section, we discuss two alternative mechanisms for generating fair summaries.

5.1 Summarizing classes separately

Suppose that the textual units in the input belong to t classes $Z_1, Z_2, \ldots, Z_t$, and to conform to a desired fairness notion, the summary should have c_i units from class Z_i, $i = 1, 2, \ldots, t$ (using the same notations as in Section 4). The easiest way to generate a fair summary is to separately summarize the textual units belonging to each class Z_i, to produce a summary of length c_i, and finally to combine all the t summaries to obtain the final summary of length k. We refer to this method as the **ClasswiseSumm** method – specifically, we use our proposed algorithm, without any fairness constraints, to summarize each class separately.

While this method is possibly the easiest to generate a fair summary, it is not clear how good the resultant summaries will be. We will evaluate this method in Section 6.

5.2 Fair ranking based summarization

Many summarization algorithms (including the unsupervised ones stated in Section 2) generate an importance score of each textual unit in the input. The textual units are then ranked in decreasing order of this importance score, and the top-ranked k units are selected to form the summary. Hence, if the ranked list of the textual units can be made fair (according to some fairness notion), then selecting the top k from this fair ranked list can be another way of generating fair summaries.

Zehlike *et al.* [34] recently proposed a fair-ranking scheme in a **two-class** setting, with a 'majority class' and a 'minority class' for which fairness has to be ensured adhering to a *ranked group fairness criterion* . They proposed a ranking algorithm *FA*IR* that ensures that the proportion of the candidates/items from the minority class in a ranked list never falls below a certain specified threshold. Zehlike *et al.* [34] ensure two fairness criteria – *selection utility* which means every selected item is more qualified than those not selected, and *ordering utility* which means for every pair of selected candidates, either the more qualified is ranked above or the difference in their qualifications is small.

Applying FA*IR algorithm for summarization: We use this algorithm for fair extractive text summarization as follows. We consider that class to be the majority class which has the higher number of textual units in the input data, while the class having lesser textual units in the input is considered the minority class.

Input: The algorithm takes as input a set of textual units (to be summarized); the other input parameters (k, q_i, g_i and p) are set as discussed below.

Parameter settings: We set the parameters of FA*IR as follows.

- *Qualification (q_i) of a candidate:* In our summarization setting, this is the goodness value of a textual unit in the data to be summarized. We set this value to the importance score computed by some standard summarization algorithm (e.g., the ones discussed in Section 2.1) that ranks the text units by their importance scores.
- *Expected size (k) of the ranking:* The expected number of textual units in the summary (k).
- *Indicator variable (g_i) indicating if the candidate is protected:* We consider that class to be the minority class which has the lesser number of textual units in the input data.
- *Minimum proportion (p) of protected candidates:* We will set this value in the open interval]0, 1[(0 and 1 excluded) so that a particular notion of fairness is ensured in the summary. For instance, if we want equal representation of both classes in the summary, we will set $p = 0.5$.
- *Adjusted significance level (α_c):* We regulate this parameter in the open interval]0, 1[.

Working of the algorithm: Two priority queues P_0 (for the textual units from the majority class) and P_1 (for the textual units of the minority class), each with capacity k, are set to empty. Then P_0 and P_1 are initialized by the goodness values (q_i) of the majority and minority textual units respectively. Then a ranked group fairness table is created which calculates the minimum number of minority items/candidates at each rank, given the parameter setting. If this table determines that a textual unit from the minority class needs to added to the summary (being generated), the algorithm adds the best element from P_1 to the summary S; otherwise it adds the overall best textual unit (from $P_0 \cup P_1$) to S.

Output: A fair summary (S) of desired length k, adhering to a particular notion of fairness.

Note that since the *FA*IR* algorithm provides fair ranking for two classes only [34], we look to apply this algorithm for summarization of data containing textual units from exactly two classes. So, we report the summarization results using this methodology only on the Claritin dataset that has two classes – Male and Female. It is an interesting future work to design a fair ranking algorithm for more than two classes, and then to use the algorithm for summarizing data from more than two classes.

6 EXPERIMENTS AND EVALUATION

We now experiment with different methodologies of generating fair summaries, over the four datasets – the two microblog datasets Claritin, USelection, and the two DUC news article datasets D0621 and D0626 (described in Section 2). For the Claritin and USelection datasets, we generate all summaries of length $k = 50$ tweets. For the two DUC datasets, we generate all summaries of length 250 words (as specified by DUC2006). Specifically, sentences are included in the summary until the length of the summary becomes ≥ 250, and the summary is truncated to 250 words while computing ROUGE scores. To evaluate the quality of summaries, we compute ROUGE-1 and ROUGE-2 Recall and F_1 scores by matching the algorithmically generated summaries with the gold standard summaries.

Method	Nos. of tweets		ROUGE-1		ROUGE-2	
	Female	Male	Recall	F_1	Recall	F_1
Whole data	2,505 (62%)	1,532 (38%)				
Without any fairness constraint						
FairSumm	37	13	0.5487	0.5457	0.1724	0.1706
Fairness: Equal representation						
FairSumm	25	25	**0.5604**	**0.5523**	**0.1877**	**0.1849**
ClasswiseSumm	25	25	0.5453	0.5383	0.1722	0.1697
Fa*ir-ClusRank	25	25	0.4333	0.4805	0.1355	0.1625
Fa*ir-DSDR	25	25	0.2846	0.4005	0.1392	0.2063
Fa*ir-LexRank	25	25	0.2900	0.3699	0.1100	0.1534
Fa*ir-LSA	25	25	0.5135	0.4928	0.1136	0.1090
Fa*ir-LUHN	25	25	0.4153	0.4290	0.1145	0.1183
Fa*ir-SumBasic	25	25	0.3144	0.4357	0.1109	0.1538
Fa*ir-SummaRNN	25	25	0.3558	0.4099	0.1257	0.1536
Fa*ir-SummaCNN	25	25	0.3558	0.4099	0.1257	0.1536
Fairness: Proportional representation						
FairSumm	31	19	**0.5719**	**0.5681**	**0.2061**	**0.2022**
ClasswiseSumm	31	19	0.5501	0.5414	0.1803	0.1731
Fa*ir-ClusRank	31	19	0.4387	0.4830	0.1332	0.1587
Fa*ir-DSDR	31	19	0.3018	0.4251	0.1451	0.2045
Fa*ir-LexRank	31	19	0.3117	0.4061	0.1153	0.1596
Fa*ir-LSA	31	19	0.5018	0.4868	0.1181	0.1146
Fa*ir-LUHN	31	19	0.4261	0.4349	0.1190	0.1215
Fa*ir-SumBasic	31	19	0.3180	0.4355	0.1163	0.1593
Fa*ir-SummaRNN	31	19	0.3405	0.3939	0.1203	0.1475
Fa*ir-SummaCNN	31	19	0.3405	0.3939	0.1203	0.1475

Table 6: Generating fair summaries of the Claritin dataset: Number of tweets of the two classes, in the whole dataset and the summaries of length 50 tweets generated by different algorithms. Also given are the ROUGE-1 and ROUGE-2 Recall and F_1 scores of each summary.

Parameter settings of algorithms: The proposed FairSumm algorithm uses a similarity function $sim(i, j)$ to measure the similarity between two textual units i and j. We experimented with the following two similarity functions:
(1) TFIDFsim – we compute TF-IDF scores for each word (unigram) in a dataset, and hence obtain a TF-IDF vector for each textual unit. The similarity $sim(i, j)$ is computed as the cosine similarity between the TF-IDF vectors of i and j.
(2) Embedsim – we obtain embeddings for the words in a dataset, either by training Word2vec [25] on the dataset, or by considering pre-trained GloVe embeddings [29]. In either way, we get an embedding (a vector of dimension 300) for each distinct word in the dataset, which is expected to capture the semantics of the word. For a given textual unit i, we obtain an embedding by taking the average embedding of all words contained in i (note that Word2vec vectors are additive in nature [25]). The similarity $sim(i, j)$ is computed as the cosine similarity between the embeddings of i and j.
We experimented with these two similarity measures and found that the performance of the FairSumm algorithm is very similar for both. Hence, we report results for the TFIDFsim similarity measure.

For the FA*IR algorithm, the value of the parameter α_c needs to be decided (see Section 5.2). We try different values of α_c in the interval [0, 1] using grid search, and finally use $\alpha_c = 0.5$ since this value obtained the best ROUGE scores on the Claritin dataset.

Method	Nos. of tweets			ROUGE-1		ROUGE-2	
	Pro-Rep	Pro-Dem	Neutral	Recall	F_1	Recall	F_1
Whole data	1,309 (62%)	658 (31%)	153 (7%)				
Without any fairness constraint							
FairSumm	34	12	4	0.3586	0.4596	0.0743	0.0913
Fairness: Equal representation							
FairSumm	17	17	16	**0.3683**	**0.4671**	**0.0781**	**0.0965**
ClasswiseSumm	16	16	18	0.3627	0.4671	0.0711	0.0879
Fairness: Proportional representation							
FairSumm	31	15	4	**0.3756**	**0.4902**	**0.0937**	**0.1164**
ClasswiseSumm	30	15	5	0.3668	0.4541	0.0810	0.1003
Fairness: No Adverse Impact							
FairSumm	29	17	4	0.3713	0.4836	0.0861	0.1024
FairSumm	30	16	4	0.3721	0.4893	0.0869	0.1093
FairSumm	31	15	4	**0.3756**	**0.4902**	**0.0937**	**0.1164**
FairSumm	31	16	3	0.3711	0.4775	0.0854	0.0956
FairSumm	32	15	3	0.3707	0.4726	0.0849	0.0936

Table 7: Generating fair summaries of the USelection dataset: Number of tweets of the three classes, in the whole dataset and the summaries of length 50 tweets generated by different algorithms. Also given are the ROUGE-1 and ROUGE-2 Recall and F_1 scores of each summary.

Results: Table 6 reports the results of fair summarization algorithms on the Claritin dataset. Specifically, we compute summaries without any fairness constraint, and considering the two fairness notions of *equal representation* and *proportional representation* (as explained in Section 3). In each case, we state the number of sentences in the summary from the two classes, and the ROUGE scores of the summary. Similarly, Table 7, Table 8 and Table 9 report the results for the USelection, D0621 and D0626 datasets respectively. The FairSumm (proposed) and ClasswiseSumm algorithms are executed over all datasets. For the two-class Claritin dataset, we also try the methodology stated in Section 5.2 where the FA*IR algorithm is used over several existing summarization algorithms such as ClusterRank, LexRank, DSDR, etc. These methodologies are denoted as Fa*ir-ClusRank, Fa*ir-LexRank, Fa*ir-DSDR, and so on.

Note that, for generating a fixed length summary, the neural SummaRuNNer model uses only the textual units labeled with 1, ranked as per their confidence scores. While applying FA*IR algorithm over SummaRuNNer (Fa*ir-SummaRNN and Fa*ir-SummaCNN in Table 6), we have given as input to FA*IR the ranked list of only those textual units that are labeled with 1. Theoretically, it may so happen that there might not be sufficient representation (required for a certain fairness notion) of a certain class in the set of textual units that are labeled with 1. In such cases, a fair ranking algorithm can not make the final summaries fair. We make the following observations from the results.

Ensuring fairness can lead to better summarization: FairSumm with fairness constraints always achieves higher ROUGE scores than FairSumm without any fairness constraints. In some cases, ClasswiseSumm with fairness constraints also achieve higher ROUGE scores than FairSumm without fairness constraints. Also, from Table 1 and Table 6 (both on Claritin dataset), we can compare the performance of the existing summarization algorithms (e.g., DSDR, LexRank) without any fairness constraint, and after their outputs are made fair using the methodology in Section 5.2. We find that the performances are comparable – while most of the ROUGE-1 Recall scores are higher in Table 6 (when the summary is made fair), most ROUGE-2 Recall scores are higher in Table 1 (in the original summaries). The trends are similar in case of F_1 Scores.

We also observe that summaries that conform to proportional representation generally achieve higher ROUGE scores than summaries that conform to equal representation.

Thus, *making summaries fair can actually improve quality of summaries* (as measured by ROUGE scores). These higher ROUGE scores of fair summaries are probably due to the human assessors inherently attempting to represent different classes of textual units in the gold standard summaries in a similar proportion as the classes occur in the input data.

Summarizing different classes separately does not yield good summarization: Across all datasets, the proposed FairSumm algorithm achieves higher ROUGE scores than the ClasswiseSumm approach, considering the same fairness notion. Note that in the ClasswiseSumm approach, the same algorithm is used on each class separately. Hence, separately summarizing each class leads to relatively poor summaries, as compared to the proposed methodology.

FairSumm generalizable to different fairness notions: The proposed FairSumm algorithm is generalizable to various fairness notions. Apart from equal representation and proportional representation, we also experimented with the 'No adverse impact' notion – the last few rows of Table 7 shows different summaries that can be generated using FairSumm considering 'no adverse impact' as the fairness notion (such rows are omitted from other tables for brevity).

Comparing FairSumm with other algorithms: As demonstrated in Table 1, Table 2, Table 3, and Table 4, none of the existing summarization algorithms generate fair summaries. For most of the datasets (except Claritin), the recently proposed supervised neural algorithms achieve higher ROUGE scores than the unsupervised algorithms. But the summaries produced by the neural methods under-represent the minority neutral group in the USelection dataset, and two of the three sources in the DUC datasets.

The summaries generated by FairSumm, apart from ensuring fairness, achieve very comparable ROUGE scores as the best-performing algorithms. For the two microblog datasets (Claritin and USelection), FairSumm with 'proportional representation' achieves higher ROUGE scores than all other methods.

For the D0626 dataset, comparing Table 4 and Table 9, we can see that the performance of FairSumm with 'proportional representation' actually surpasses that of all algorithms, including the neural models and all the summaries submitted in the DUC2006 track. For the topic D0621, we can see from Table 3 and Table 8 that the performance of FairSumm with proportional representation is better than that of all methods (including neural models) except the best submitted summary in the DUC track (Peer 24).

Lastly, across all the datasets, the summaries generated by the proposed FairSumm algorithm achieve the highest ROUGE scores, compared to all other methods for generating fair summaries.

Method	Nos. of tweets			ROUGE-1		ROUGE-2	
	APW	NYT	XIE	Recall	F_1	Recall	F_1
Whole data	167 (44%)	78 (21%)	131 (35%)				
Without any fairness constraint							
FairSumm	4	1	3	0.4141	0.3972	0.0714	0.0722
Fairness: Equal representation							
FairSumm	3	3	3	**0.4374**	**0.4146**	**0.0881**	**0.0722**
ClasswiseSumm	4	3	3	0.4220	0.4004	0.0737	0.0752
Fairness: Proportional representation							
FairSumm	4	2	3	**0.4691**	**0.4643**	**0.0917**	**0.0893**
ClasswiseSumm	4	1	3	0.4423	0.4393	0.0804	0.0796

Table 8: Generating fair summaries of DUC06, topic D0621 dataset: Number of sentences from the three sources, in the whole dataset and the summaries of length 250 words generated by different algorithms. Also given are the ROUGE-1 and ROUGE-2 Recall and F_1scores of each summary.

Method	Nos. of tweets			ROUGE-1		ROUGE-2	
	APW	NYT	XIE	Recall	F_1	Recall	F_1
Whole data	338 (69%)	89 (18%)	65 (13%)				
Without any fairness constraint							
FairSumm	5	1	2	0.4686	0.4021	0.1332	0.1302
Fairness: Equal representation							
FairSumm	3	3	3	**0.4743**	**0.4370**	**0.1245**	**0.1146**
ClasswiseSumm	3	3	3	0.4511	0.3887	0.1183	0.1018
Fairness: Proportional representation							
FairSumm	6	2	1	**0.4786**	**0.4270**	**0.1802**	**0.1514**
ClasswiseSumm	5	2	2	0.4623	0.3892	0.1204	0.1121

Table 9: Generating fair summaries of DUC06, topic D0626 dataset: Number of sentences from the three sources, in the whole dataset and the summaries of length 250 words generated by different algorithms, and the ROUGE-1 and ROUGE-2 Recall and F_1 scores of each summary.

Overall, the results signify that, FairSumm can not only ensure various fairness notions in the summaries, but also can generate summaries that achieve comparable (or better) ROUGE scores than many well-known summarization algorithms.

7 RELATED WORKS

The expanding availability of textual information has demanded exhaustive research in the area of automatic text summarization. A large number of text summarization algorithms have been proposed; the reader can refer to [15] for surveys. One of the most commonly used class of summarization algorithms is centered around the popular TF-IDF model [31]. Different works have used TF-IDF score based similarities for summarization [1, 30]. Additionally, there has been a series of works where summarization has been treated as a sub-modular optimization framework [3, 21]. The algorithm proposed in this work is also based on a sub-modular constrained optimization framework, and uses the notion of TF-IDF similarity .

Given that information filtering algorithms have far-reaching social and economic consequences in today's world, fairness and anti-discrimination have been recent inclusions in the algorithm design perspective [16]. There have been several recent works on measuring different notions of fairness and biases [5, 7, 28] as well as on removing the existing unfairness from different methodologies[17, 35]. Different fairness-aware algorithms have been proposed to achieve group and/or individual fairness for tasks such as clustering [8], classification [33], ranking [34], sampling [6], and so on. However, to the best of our knowledge, there has not been prior exploration of fairness in the context of summarization.

8 CONCLUSION

To our knowledge, this work is the first attempt to develop a fairness-preserving text summarization algorithm. Through experiments on microblog datasets and the popular DUC datasets, we show that existing algorithms often produce summaries that are not fair. The proposed algorithm can generate high-quality summaries that conform to various standard notions of fairness. In fact, ensuring the fairness of summaries often leads to enhancing the quality of the summary as well.

The proposed algorithm will help in addressing the concern that using a (inadvertently) 'biased' summarization algorithm can reduce the visibility of the voice/opinion of a certain social group or source in the summary. Moreover, downstream applications that use the summaries (e.g., for opinion classification and rating inference [22]) would benefit from a fair summary (e.g., that fairly represents the positive and negative opinions in the input).

We believe that this work will open up interesting research problems on fair summarization, such as extending the concept of fairness to abstractive summaries, estimating user preferences for fair summaries in various applications, and so on.

REFERENCES

[1] Rasim M Alguliev, Ramiz M Aliguliyev, Makrufa S Hajirahimova, and Chingiz A Mehdiyev. 2011. MCMR: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications* 38, 12 (2011).

[2] Mehdi Allahyari, Seyed Amin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. 2017. Text Summarization Techniques: A Brief Survey. *CoRR* abs/1707.02268 (2017). http://arxiv.org/abs/1707.02268

[3] Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. 2014. Streaming submodular maximization: Massive data summarization on the fly. In *ACM KDD.*

[4] Dan Biddle. 2006. *Adverse Impact and Test Validation: A Practitioner's Guide to Valid and Defensible Employment Testing.* Routledge.

[5] Francesco Bonchi, Sara Hajian, Bud Mishra, and Daniele Ramazzotti. 2017. Exposing the probabilistic causal structure of discrimination. *International Journal of Data Science and Analytics* 3, 1 (2017).

[6] L Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. 2016. How to be Fair and Diverse? *arXiv preprint arXiv:1610.07183* (2016).

[7] Abhijnan Chakraborty, Johnnatan Messias, Fabricio Benevenuto, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2017. Who makes trends? understanding demographic biases in crowdsourced recommendations.

[8] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair Clustering Through Fairlets. In *NIPS.*

[9] K. Darwish, W. Magdy, and Zanouda T. 2017. Trump vs. Hillary: What Went Viral During the 2016 US Presidential Election. In *SocInfo.*

[10] Yue Dong. 2018. A Survey on Neural Network-Based Summarization Methods. *arXiv preprint arXiv:1804.04589* (2018).

[11] Nan Du, Yingyu Liang, Maria-Florina Balcan, and Le Song. 2013. Continuous-Time Influence Maximization for Multiple Items. *CoRR* abs/1312.2164 (2013). arXiv:1312.2164 http://arxiv.org/abs/1312.2164

[12] Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *J. Artif. Int. Res.* 22, 1 (2004).

[13] Nikhil Garg and others. 2009. Clusterrank: a graph based method for meeting summarization. In *INTERSPEECH.*

[14] Yihong Gong and Xin Liu. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *ACM SIGIR*.

[15] Vishal Gupta and Gurpreet Singh Lehal. 2010. A Survey of Text Summarization Extractive Techniques. *IEEE Journal of Emerging Technologies in Web Intelligence* 2, 3 (2010).

[16] Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *ACM KDD*.

[17] Sara Hajian, Josep Domingo-Ferrer, and Oriol Farràs. 2014. Generalization-based privacy preservation and discrimination prevention in data publishing and mining. *Data Mining and Knowledge Discovery* 28 (2014).

[18] Zhanying He and others. 2012. Document Summarization Based on Data Reconstruction. In *AAAI*.

[19] Andreas Krause and Daniel Golovin. 2014. Submodular Function Maximization. In *Tractability: Practical Approaches to Hard Problems*.

[20] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proc. Workshop on Text Summarization Branches Out, ACL*.

[21] Hui Lin and Jeff Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In *ACL (HLT '11)*.

[22] Elena Lloret, Horacio Saggion, and Manuel Palomar. 2010. Experiments on summary-based opinion classification. In *NAACL HLT*.

[23] H. P. Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM J. Res. Dev.* 2, 2 (1958).

[24] Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. 2011. k-NN as an implementation of situation testing for discrimination discovery and prevention. In *ACM KDD*.

[25] T. Mikolov, W.T. Yih, and G. Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *NAACL HLT*.

[26] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents.. In *AAAI*. 3075–3081.

[27] Ani Nenkova and Lucy Vanderwende. 2005. *The impact of frequency on summarization.* Technical Report. Microsoft Research.

[28] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *ACM KDD*.

[29] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proc. EMNLP*.

[30] Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the Special Issue on Summarization. *Comput. Linguist.* 28, 4 (2002).

[31] Gerard Salton. 1989. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley* (1989).

[32] Anurag Shandilya, Kripabandhu Ghosh, and Saptarshi Ghosh. 2018. Fairness of Extractive Text Summarization. In *Proc. The Web Conference (WWW) Companion Volume*. 97–98.

[33] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *AIStats*.

[34] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *ACM CIKM*.

[35] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *ICML*.