Quantifying Uncertainty in context of Natural Language Processing

by

Taehee Jung

B.A in Management Science, KAIST, 2011

M.A in Statistics, University of California, Berkeley, 2017

Submitted to the Graduate Faculty of

the Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Taehee Jung

It was defended on

July 21, 2022

and approved by

Lucas K. Mentch, PhD, Department of Statistics

Yu Cheng, PhD, Department of Statistics

Kehui Chen, PhD, Department of Statistics

Meredith L Wallace, PhD, Department of Psychiatry

Quantifying Uncertainty in context of Natural Language Processing

Taehee Jung, PhD

University of Pittsburgh, 2022

Despite recent advances in statistical machine learning that significantly improve performance, the uncertainty behind models remains largely underexplored. We identify two sources of uncertainty in this dissertation, one coming from learning sources such as algorithms or datasets and the other from the model's predicted output. In order to better understand or even improve the model's results, we then quantify two uncertainties. In particular, we study three topics of uncertainty quantification in the context of natural language processing (NLP). Firstly, we quantify model and corpus biases in text summarization based on three sub-aspects; position, importance, and diversity. Secondly, we develop a simple but effective end-to-end procedure for improving the performance of text classification tasks and the quality of the model calibration. Finally, we propose a new framework of model calibration to interpret individual point estimations with confidence and show less-biased relative frequency approximation in classification.

Keywords: Model Uncertainty, Model Calibration, Confidence Interval, Natural Language Processing, Text Summarization, Text Classification.

Table of Contents

List of Tables

List of Figures

Preface

This thesis would not have been possible without the help and support of my committee members, collaborators, and cohorts.

First of all, I would like to thank my "coolest" advisor, Dr. Lucas Mentch. I might be among the most challenging students when I think back of my Ph.D. life. Sometimes I was rough and I just failed. He got me out of trouble every time I got into it. He made Dr. Taehee Jung, and I will never forget his support.

I was pleased to have committee members, Dr. Yu Cheng, Dr. Kehui Chen, and Dr. Meredith Wallace. Your advice and comments enrich my dissertation. In particular, I appreciate Dr. Cheng and Dr. Chen for advising me to have another chance to stay in the Ph.D. program after I failed the qualifying exam. Your care and support keep me on track with the program. I want to thank my former committee member, Dr. Satish Iyengar, for his help and support, and I wish him the best of health.

During my Ph.D., I had a chance to work with mentors who always supported my work and encouraged me to complete research I have been interested in. Dr. Thomas Schaaf, I appreciate all his mentoring and especially made me aware of the topic of calibration for my dissertation. Thank you to Dr. Sungjin Lee and Dr. Joo-Kyung Kim who helped me get a job offer after I graduate. I also thank Dr. Tommy Powers and Liyuan Lee, who supervised me during my first internship at Amazon Alexa AI.

I could not complete the Ph.D. study without these excellent cohorts. Thanks, Siyu Zhou, Huy Le, Marc Richards, and Manuel Garcia, for working and studying with me. Haeun Moon and Jinwoo Cho, two other Korean cohorts who were like family during my time living in Pittsburgh, are also appreciated. Jooyeon Woo and Eunsol Kim, my lifetime best friends living in the U.S., thank you for becoming my mental supporters. I could recover whenever I talked or spent some time with them.

My family was the biggest supporter of my Ph.D. life. My parents, Moyses and Lucia, I am so proud of myself that I am your daughter. The woman who raised me with my parents, Aunt Catharina, deserves special thanks. My love for you is unconditional, and I will do

whatever I can to keep you healthy.

Last but not least, I would not start a whole new journey in my life without my partner Dongyeop Kang. You are just beyond love. I refer to you as my partner, my friend, my only family in the U.S., and my most prominent collaborator. I am so excited you will always be with me wherever I go in the future. My graduate work should be dedicated to you.

1.0 Introduction

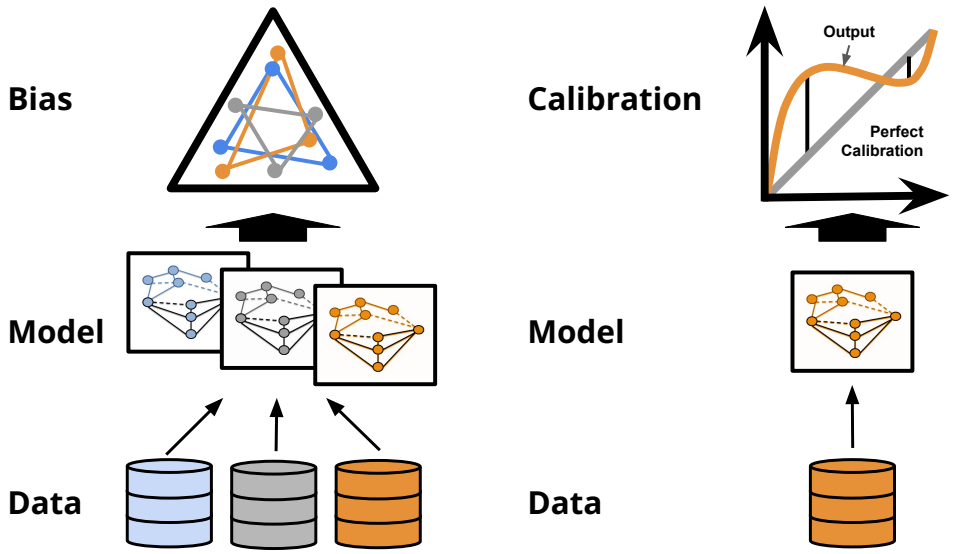
Uncertainty quantification is the study of mathematically characterizing and reducing uncertainties for any computational and real world applications. Specifically for statistical machine learning models, defining the source of uncertainties and quantifying them is an important as it helps improve the model performance and better understand the result correctly. For example, Kennedy and O'Hagan [2001] categorize the source of uncertainty of computer models into six groups, such as parameter uncertainty, model inadequacy, and residual variability and so on. However, how to define the source of uncertainties can vary according to the purpose of study.

In this work, we simply consider two sources of uncertainties, the uncertainty by the learning source and the uncertainty of predicted outputs as described in Figure 1 and show how to quantify them. In particular, we explore three topics of uncertainty quantification, mainly on natural language processing tasks.

In Chapter 2, we first explore the uncertainty by learning source, such as model algorithm and/or datasets in text summarization. Here we define three sub-aspects for text summarization; position, importance, and diversity and analyze how existing corpora and models are biased toward certain aspects differently. We find that news articles tend to be summarized with the first few sentences (position) while academic papers consider more about a coverage of contents (diversity). In addition, neural models are well-balanced on sub-aspects by yielding a better performance. Understanding such biases on model and corpus plays an important role to discover a source of uncertainties in the text summarization.

We then focus on the model uncertainty of predicted outputs or confidence, which is called "model calibration" in Chatper 3 and 4. Here, a model calibration generally refers to the study to show how predicted probabilities align to the actual relative frequencies in classification.

Specifically in Chapter 3, we propose a new method to improve model calibration in text classification. Unlike previous methods which are post-hoc and can not improve a task performance, our method directly applies an auxiliary loss on the classical objective (e.g.,



(a) Uncertainty by learning source (b) Uncertainty of model outputs

Figure 1: Description of two source of uncertainties we explore in this work.

cross-entropy) while training models and improves both the quality of model calibration and the task performance at the end. We conduct an extensive experiments over a variety of text classification tasks (e.g., ≥ 20) and show our method consistently works on them.

In Chapter 4, we first emphasize that for classification task, current metric to quantify model calibration which is a simple statistic, can not capture the level of uncertainty of a single relative frequency estimation, as well as the estimation itself is ad-hoc. We propose a less-biased method for the relative frequency approximation using a simple nearest neighbor method and provide an interval-based evidence to interpret them properly. Our work show a new framework of model calibration for an individual point, which we call customized calibration, thus, can be of tremendous practical use of human decision makers, especially for the sensitive cases such as a recidivism or a medical diagnosis.

2.0 Quantifying Corpus and System Bias in Text Summarization

Despite the recent developments on neural summarization systems, the underlying logic behind the improvements from the systems and its corpus-dependency remains largely unexplored. Position of sentences in the original text, for example, is a well known bias for news summarization. Following in the spirit of the claim that summarization is a combination of sub-functions, we define three sub-aspects of summarization: `position`, `importance`, and `diversity` and conduct an extensive analysis of the biases of each sub-aspect with respect to the domain of nine different summarization corpora (e.g., news, academic papers, meeting minutes, movie script, books, posts). We find that while `position` exhibits substantial bias in news articles, this is not the case, for example, with academic papers and meeting minutes. Furthermore, our empirical study shows that different types of summarization systems (e.g., neural-based) are composed of different degrees of the sub-aspects. Our study provides useful lessons regarding consideration of underlying sub-aspects when collecting a new summarization dataset or developing a new system. The following sections are mainly from Jung et al. [2019].

2.1 Introduction

Despite numerous recent developments in neural summarization systems [Narayan et al., 2018b, Nallapati et al., 2016, See et al., 2017, Kedzie et al., 2018, Gehrmann et al., 2018, Paulus et al., 2017] the underlying rationales behind the improvements and their dependence on the training corpus remain largely unexplored. Edmundson [1969] put forth the position hypothesis: important sentences appear in preferred positions in the document. Lin and Hovy [1997] provide a method to empirically identify such positions. Later, Hong and Nenkova [2014] showed an intentional lead bias in news writing, suggesting that sentences appearing early in news articles are more important for summarization tasks. More generally, it is well known that recent state-of-the-art models [Nallapati et al., 2016, See et al., 2017] are often

marginally better than the first-k baseline on single-document news summarization.

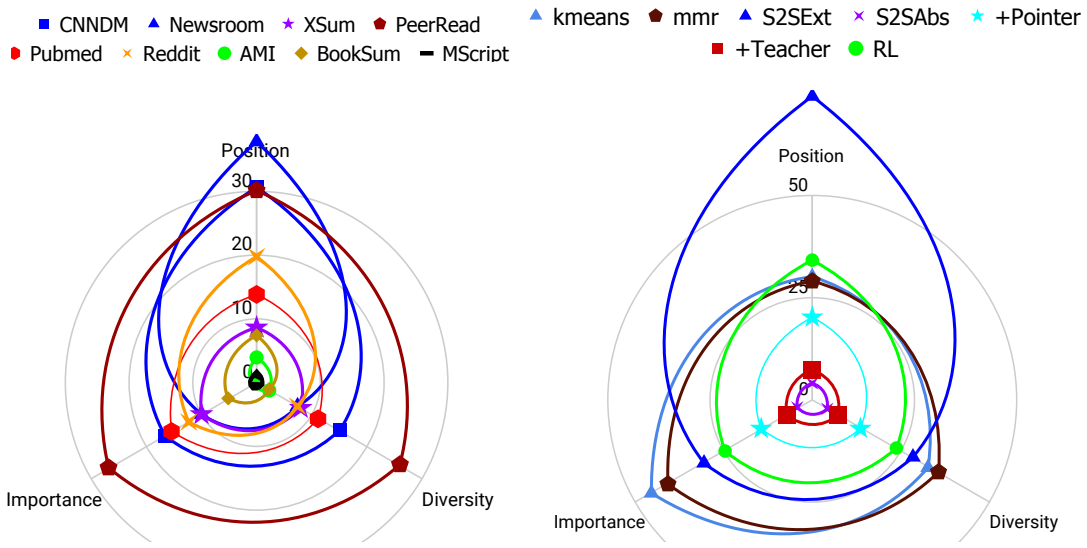
In order to address the position bias of news articles, Narayan et al. [2018a] collected a new dataset called XSum to create single sentence summaries that include material from multiple positions in the source document. Kedzie et al. [2018] showed that the position bias in news articles is not the same across other domains such as meeting minutes [Carletta et al., 2005].

In addition to `position`, Lin and Bilmes [2012] defined other sub-aspect functions of summarization including `coverage`, `diversity`, and `information`. Lin and Bilmes [2011] claim that many existing summarization systems are instances of mixtures of such sub-aspect functions; for example, maximum marginal relevance (MMR) [Carbonell and Goldstein, 1998] can be seen as an combination of diversity and importance functions.

Following the sub-aspect theory, we explore three important aspects of summarization (§2.3): **position** for choosing sentences by their position, **importance** for choosing relevant contents, and **diversity** for ensuring minimal redundancy between summary sentences.

We then conduct an in-depth analysis of these aspects over nine different domains of summarization corpora (§2.5) including news articles, meeting minutes, books, movie scripts, academic papers, and personal posts. For each corpus, we investigate which aspects are most important and develop a notion of **corpus bias** (§2.6). We provide an empirical result showing how current summarization systems are compounded of which sub-aspect factors called **system bias** (§2.7). At last, we summarize our actionable messages for future summarization researches (§2.8). We summarize some notable findings as follows:

- Summarization of personal post and news articles except for XSum [Narayan et al., 2018a] are biased to the position aspect, while academic papers are well balanced among the three aspects (see Figure 2 (a)). Summarizing long documents (e.g. books and movie scripts) and conversations (e.g. meeting minutes) are extremely difficult tasks that require multiples aspects together.
- Biases do exist in current summarization systems (Figure 2 (b)). Simple ensembling of multiple aspects of systems show comparable performance with simple single-aspect systems.
- Reference summaries in current corpora include less than 15% of new words that do not



(a) Corpus bias (b) System bias

Figure 2: Corpus and system biases with the three sub-aspects, showing what portion of aspect is used for each corpus and each system. The portion is measured by calculating ROUGE score between (a) summaries obtained from each aspect and target summaries or (b) summaries obtained from each aspect and each system. For system bias, we show for `CNNDM`. Other corpora are in Appendix E.

appear in the source document, except for abstract text of academic papers.

- Semantic volume [Yogatama et al., 2015] overlap between the reference and model summaries is not correlated with the hard evaluation metrics such as ROUGE [Lin, 2004].

2.2 Related Work

We provide here a brief review of prior work on summarization biases. Lin and Hovy [1997] studied the position hypothesis, especially in the news article writing [Hong and Nenkova, 2014, Narayan et al., 2018a] but not in other domains such as conversations [Kedzie et al., 2018]. Narayan et al. [2018a] collected a new corpus to address the bias by compressing

multiple contents of source document in the single target summary. In the bias analysis of systems, Lin and Bilmes [2012, 2011] studied the sub-aspect hypothesis of summarization systems. Our study extends the hypothesis to various corpora as well as systems. With a specific focus on `importance` aspect, a recent work [Peyrard, 2019a] divided it into three sub-categories; redundancy, relevance, and informativeness, and provided quantities of each to measure. Compared to this, ours provide broader scale of sub-aspect analysis across various corpora and systems.

We analyze the sub-aspects on different domains of summarization corpora: news articles [Nallapati et al., 2016, Grusky et al., 2018, Narayan et al., 2018a], academic papers or journals [Kang et al., 2018, Kedzie et al., 2018], movie scripts [Gorinski and Lapata, 2015], books [Mihalcea and Ceylan, 2007], personal posts [Ouyang et al., 2017], and meeting minutes [Carletta et al., 2005] as described further in §2.5.

Beyond the corpora themselves, a variety of summarization systems have been developed: [Mihalcea and Tarau, 2004, Erkan and Radev, 2004] used graph-based keyword ranking algorithms. [Lin and Bilmes, 2010, Carbonell and Goldstein, 1998] found summary sentences which are highly relevant but less redundant. Yogatama et al. [2015] used semantic volumes of bigram features for extractive summarization. Internal structures of documents have been used in summarization: syntactic parse trees [Woodsend and Lapata, 2011, Cohn and Lapata, 2008], topics [Zajic et al., 2004, Lin and Hovy, 2000], semantic word graphs [Mehdad et al., 2014, Gerani et al., 2014, Ganesan et al., 2010, Filippova, 2010, Boudin and Morin, 2013], and abstract meaning representation [Liu et al., 2015]. Concept-based Integer-Linear Programming (ILP) solver [McDonald, 2007] is used for optimizing the summarization problem [Gillick and Favre, 2009, Banerjee et al., 2015, Boudin et al., 2015, Berg-Kirkpatrick et al., 2011]. Durrett et al. [2016] optimized the problem with grammatical and anarphorcity constraints.

With a large scale of corpora for training, neural network based systems have recently been developed. In abstractive systems, Rush et al. [2015] proposed a local attention-based sequence-to-sequence model. On top of the seq2seq framework, many other variants have been studied using convolutional networks [Cheng and Lapata, 2016, Allamanis et al., 2016], pointer networks [See et al., 2017], scheduled sampling [Bengio et al., 2015], and reinforce-

ment learning [Paulus et al., 2017]. In extractive systems, different types of encoders [Cheng and Lapata, 2016, Nallapati et al., 2017, Kedzie et al., 2018] and optimization techniques [Narayan et al., 2018b] have been developed. Our goal is to explore which types of systems learns which sub-aspect of summarization.

2.3 Sub-aspects of Summarization

We focus on three crucial aspects : **Position**, **Diversity**, and **Importance**. For each aspect, we use different extractive algorithms to **capture how much of the aspect is used in the oracle extractive summaries**[1]. For each algorithm, the goal is to select k extractive summary sentences (equal to the number of sentences in the target summaries for each sample) out of N sentences appearing in the original source. The chosen sentences or their indices will be used to calculate the various evaluation metrics described in §2.4

For some algorithms below, we use vector representation of sentences. We parse a document x into a sequence of sentences $x = x_1..x_N$ where each sentence consists of a sequence of words $x_i = w_1..w_s$. Each sentence is then encoded:

$$\mathbf{E}(x_i) = \texttt{BERT}(w_{i,1}..w_{i,s}) \tag{1}$$

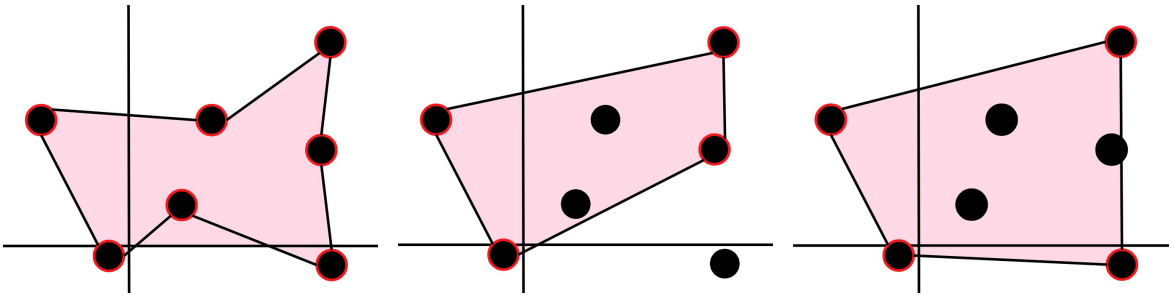
where `BERT` [Devlin et al., 2019] is a pre-trained bidirectional encoder from transformers [Vaswani et al., 2017][2]. We use the last layer from `BERT` as a representation of each token, and then average them to get final representation of a sentence. All tokens are lower cased.

2.3.1 Position

Position of sentences in the source has been suggested as a good indicator for choosing summary sentences, especially in news articles [Lin and Hovy, 1997, Hong and Nenkova, 2014, See et al., 2017]. We compare three position-based algorithms: **First**, **Last**, and **Middle**, by simply choosing k number of sentences in the source document from these positions.

[1] See §2.4 for our oracle set construction.

[2] The other encoders such as averaging word embeddings [Pennington et al., 2014] show comparable performance.



(a) Default (b) Heuristic (c) ConvexFall

Figure 3: Volume maximization functions. Black dots are sentences in source document, and red dots are chosen summary sentences. The red-shaded polygons are volume space of the summary sentences.

2.3.2 Diversity

[Yogatama et al., 2015] assume that extractive summary sentences which maximize the semantic volume in a distributed semantic space are the most diverse but least redundant sentences. Motivated by this notion, our goal is to find a set of k sentences that maximizes the volume size of them in a continuous embedding space like the BERT representations in Eq 1. Our objective is to find the optimal search function $\mathcal{S}$ that maximizes the volume size $\mathcal{V}$ of searched sentences: $\arg\max_{1..k} \mathcal{V}\left(\mathcal{S}_{1..c}\left(\mathbf{E}(x_1), \ldots, \mathbf{E}(x_N)\right)\right)$.

If k=N, we use every sentence from the source document. (Figure 3 (a)). However, its volume space does not guarantee to maximize the volume size because of the non-convex polygonality. In order to find a convex maximum volume, we consider two different algorithms described below.

Heuristic. [Yogatama et al., 2015] heuristically choose a set of summary sentences using a greedy algorithm: It first chooses a sentence which has the farthest vector representation from the centroid of whole source sentences, and then repeatedly finds sentences whose representation is farthest from the centroid of vector representations of the chosen sentences. Unlike the original algorithm in [Yogatama et al., 2015] restricting the number of words, we constrain the total number of selected sentences to k. This heuristic algorithm can fail to find the maximum volume depending on its starting point and/or the farther distance between

two points detected (Figure 3 (b)).

ConvexFall. Here we first find the convexhull[3] using Quickhull [Barber et al., 1996], implemented by Qhull library[4]. It guarantees the maximum volume size of selected points with minimum number of points (Figure 3 (c)). However, it does not reduce a redundancy between the points over the convex-hull, and usually choose larger number of sentences than k. Marcu [1999] shows an interesting study regarding an importance of sentences: given a document, if one deletes the least central sentence from the source text, then at some point the similarity with the reference text rapidly drops at sudden called the *waterfall* phenomena. Motivated by his study, we similarly prune redundant sentences from the set chosen by convex-hull search. For each turn, the sentence with the lowest volume reduction ratio is pruned until the number of remaining sentences is equivalent to k.

2.3.3 Importance

We assume that contents that repeatedly occur in one document contain *important* information. We find sentences that are nearest to the neighbour sentences using two distance measures: **N-Nearest** calculates an averaged Pearson correlation between one and the rest for all source sentence vector representations. k sentences having the highest averaged correlation are selected as final extractive summaries. On the other hand, **K-Nearest** chooses the K nearest sentences per each sentence, and then averages distances between each nearest sentence and the selected one. The one has the lowest averaged distance is chosen. This calculation is repeated k times and the selected sentences are removed from the remaining pool.

2.4 Metrics

In order to determine the aspects most crucial to the summarization task, we use three evaluation metrics:

[3]Definition: a set of points is defined as the smallest convex set that includes the points.
[4]http://www.qhull.org/

`ROUGE` is Recall-Oriented Understudy for Gisting Evaluation [Lin and Hovy, 2000] for evaluating summarization systems. We use ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) F-measure scores which corresponds to uni-gram, bigrams and longest common subsequences, respectively, and their averaged score (R).

`Volume Overlap (VO)` ratio. Hard metrics like ROUGE often ignore semantic similarities between sentences. Based on the volume assumption in Yogatama et al. [2015], we measure overlap ratio of two semantic volumes calculated by the model and target summaries. We obtain a set of vector representations of the reference summary sentences $\hat{Y}$ and the model summary sentences Y predicted by any algorithm *algo* in §2.3 for the i-th document:

$$\hat{Y}_i = (\, \hat{y}_{i,1} \,..\, \hat{y}_{i,k} \,), \quad Y_i^{algo} = (\, y_{i,1}^{algo} \,..\, y_{i,k}^{algo} \,) \tag{2}$$

Each volume `V` is then calculated using the convex-hull algorithm and their overlap ($\sqcap$) is calculated using a shapely package[5][6]. The final `VO` is then:

$$\mathbf{VO}_{algo} = \sum_{i=1}^{N} \frac{\mathtt{V}(\, E(Y_i^{algo})) \sqcap \mathtt{V}\,(E(\hat{Y}_i))}{\mathtt{V}\,(E(\hat{Y}_i))} \tag{3}$$

where N is the total number of input documents, E is the BERT sentence encoder in Eq 1, and $E(\hat{Y}_i)$ and $E(Y_i^{algo})$ are a set of vector representations of the reference and model summary sentences, respectively. The volume overlap indicates how two summaries are semantically overlapped in a continuous embedding space.

`Sentence Overlap (SO)` ratio. Even though ROUGE provides a recall-oriented lexical overlap, we don't know the upper-bound on performance (called `oracle`) of the extractive summarization. We extract the oracle extractive sentences (i.e. a set of input sentences) which maximizes ROUGE-L F-measure score with the reference summary. We then measure sentence overlap (`SO`) which determines how many extractive sentences from our algorithms are in the oracle summary. The `SO` is:

$$\mathbf{SO}_{algo} = \sum_{i=1}^{n} \frac{\mathtt{C}(Y_i^{algo} \cap \hat{Y}_i)}{\mathtt{C}(\hat{Y}_i)} \tag{4}$$

[5] https://pypi.org/project/Shapely/

[6] Due to the lack of overlap calculation between two polygons of high dimensions, we reduce it to 2D PCA space.

	CNNDM	Newsroom	Xsum	PeerRead	PubMed	Reddit	AMI	BookSum	MScript
Source	News	News	News	Papers	Papers	Post	Minutes	Books	Script
Multi-sents.	✓	✓	X	✓	✓	X	✓	✓	✓
Data size	287K/11K	992K/109K	203K/11K	10K/550	21K/2.5K	404/48	98/20	- /53	- /1K
Avg src sents.	40/34	24/24	33/33	45/45	97/97	19/15	767/761	- /6.7K	- /3K
Avg tgt sents.	4/4	1.4/1.4	1/1	6/6	10/10	1/1	17/17	- /336	- /5
Avg src tokens	792/779	769 /762	440/442	1K/1K	2.4K/2.3K	296/236	6.1K/6.4K	- /117K	- /23.4K
Avg tgt tokens	55/58	30/31	23/23	144/146	258/258	24/25	281/277	- /6.6K	- /104

Table 1: Data statistics on summarization corpora. Source is the domain of dataset. Multi-sents. is whether the summaries are multiple sentences or not. All statistics are divided by Train/Test except for `BookSum` and `MScript`.

where `C` is a function for counting the number of elements in a set. The sentence overlap indicates how well the algorithm finds the oracle summaries for extractive summarization.

2.5 Summarization Corpora

We use various domains of summarization datasets to conduct the bias analysis across corpora and systems. Each dataset has source documents and corresponding abstractive target summaries. We provide a list of datasets used along with a brief description and our pre-processing scheme:

- `CNNDM` [Nallapati et al., 2016]: contains 300K number of online news articles. It has multiple sentences (4.0 on average) as a summary.
- `Newsroom` [Grusky et al., 2018]: contains 1.3M news articles and written summaries by authors and editors from 1998 to 2017. It has both extractive and abstractive summaries.
- `XSum` [Narayan et al., 2018a]: has news articles and their single but abstractive sentence summaries mostly written by the original author.

- `PeerRead` [Kang et al., 2018]: consists of scientific paper drafts in top-tier computer science venues as well as `arxiv.org`. We use full text of introduction section as source document and of abstract section as target summaries.
- `PubMed` [Kedzie et al., 2018]: is 25,000 medical journal papers from the PubMed Open Access Subset.[7] Unlike `PeerRead`, full paper except for abstract is used as source documents.
- `MScript` [Gorinski and Lapata, 2015]: is a collection of movie scripts from ScriptBase corpus and their corresponding user summaries of the movies.
- `BookSum` [Mihalcea and Ceylan, 2007]: is a dataset of classic books paired to summaries from Grade Saver[8] and Cliff's Notes[9]. Due to a large number of sentences, we only choose the first 1K sentences for source document and the first 50 sentences for target summaries.
- `Reddit` [Ouyang et al., 2017]: is a collection of personal posts from `reddit.com`. We use a single abstractive summary per post. The same data split from Kedzie et al. [2018] is used.
- `AMI` [Carletta et al., 2005]: is documented meeting minutes from a hundred hours of recordings and their abstractive summaries.

Table 1 summarizes the characteristics of each dataset. We note that the Gigaword [Graff et al., 2003], New York Times[10], and Document Understanding Conference (DUC)[11] are also popular datasets commonly used in summarization analyses, though here we exclude them as they represent only additional collections of news articles, showing similar tendencies to the other news datasets such as `CNNDM`.

2.6 Analysis on Corpus Bias

We conduct different analyses of how each corpus is biased with respect to the sub-aspects. We highlight some key findings for each sub-section.

[7] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/
[8] http://www.gradesaver.com
[9] http://www.cliffsnotes.com/
[10] https://catalog.ldc.upenn.edu/LDC2008T19
[11] http://duc.nist.gov

		CNNDM			NewsRoom			XSum			PeerRead			PubMed			Reddit			AMI			BookSum			MScript		
		R	VO	SO	**R**	VO	SO	**R**	VO	SO	**R**	VO	SO	**R**	VO	SO	**R**	VO	SO	**R**	VO	SO	**R**	VO	SO	**R**	VO	SO
	Random	19.1	18.6	14.6	10.1	2.1	9.0	9.3	-	8.4	27.9	42.5	26.2	30.1	46.9	13.0	11.8	-	11.3	12.0	39.3	2.4	29.4	85.8	4.9	8.1	25.2	0.1
	Oracle	42.8	-	-	48.1	-	-	19.6	-	-	46.3	-	-	47.0	-	-	30.0	-	-	32.0	-	-	38.9	-	-	24.2	-	-
Position	First	**30.7**	13.1	**30.7**	**32.2**	**4.4**	**37.8**	9.1	-	8.7	**32.0**	40.7	**30.3**	27.6	44.3	13.8	**15.3**	-	**19.9**	11.4	48.0	**3.8**	**29.1**	85.1	**7.4**	6.9	12.4	**0.7**
	Last	16.4	18.6	8.2	7.7	1.9	4.4	8.3	-	7.0	28.9	38.5	27.0	28.9	45.2	14.0	11.2	-	10.7	7.8	42.1	2.0	26.5	85.3	3.3	**8.8**	19.5	0.2
	Middle	21.5	18.7	11.8	12.4	1.9	5.6	9.1	-	9.1	29.7	40.7	22.8	28.9	45.9	12.3	11.5	-	7.1	11.1	36.4	2.3	27.9	83.0	4.9	8.0	23.9	0.1
Divers.	ConvFall	21.6	**57.7**	15.0	10.6	4.2	7.3	8.4	-	8.0	29.8	**77.5**	25.9	28.2	**93.5**	11.2	11.6	-	7.5	**14.0**	**98.6**	2.4	16.9	**99.7**	2.2	8.5	**59.2**	0.2
	Heuris.	21.4	19.8	14.6	10.5	2.4	7.6	8.4	-	8.1	29.2	36.6	24.8	27.5	59.7	10.5	11.5	-	7.1	10.7	66.0	2.4	26.9	99.7	4.5	6.4	5.7	0.2
Import.	NNear.	22.0	3.3	16.6	13.5	0.5	10.0	**9.8**	-	**10.1**	30.6	8.4	26.7	**31.8**	9.3	**15.5**	13.8	-	12.2	1.3	0.2	0.1	27.9	1.5	5.1	8.7	0.9	0.3
	KNear.	23.0	3.9	17.7	14.0	0.7	10.9	9.3	-	9.1	30.6	9.9	27.0	29.6	10.5	15.0	10.4	-	8.5	0.0	0.1	0.0	21.8	1.4	3.7	0.6	0.0	0.1

Table 2: Comparison of different corpora w.r.t the three sub-aspects: **position**, **diversity**, and **importance**. We averaged R1, R2, and RL as **R** (See Appendix C for full scores). Note that volume overlap (VO) doesn't exist when target summary has a single sentence. (i.e., XSum, Reddit)

2.6.1 Multi-aspect analysis

Table 2 shows a comparison of the three aspects for each corpus where we include random selection and the oracle set. For each dataset metrics are calculated on a test set except for BookSum and AMI where we use train+test due to the smaller sample size.

Earlier isn't always better. Sentences selected early in the source show high ROUGE and SO on CNNDM, Newsroom, Reddit, and BookSum, but not in other domains such as medial journals and meeting minutes, and the condensed news summaries (XSum). For summarization of movie scripts in particular, the last sentences seem to provide more important summaries.

XSum **requires more importance than other corpora.** Interestingly, the most powerful algorithm for XSum is N-Nearest. This shows that summaries in XSum are indeed collected by abstracting multiple important contents into single sentence, avoiding the position bias.

First, ConvexFall and N-Nearest tend to work better than the other algorithms for each aspect. First is better than Last or Middle in new articles except for XSum and personal posts, while not in academic papers (i.e., PeerRead, PubMed) and meeting

minutes. ConvexFall finds the set of sentences that maximize the semantic volume overlap with the target sentences better than the heuristic one.

ROUGE and SO show similar behavior, while VO does not. In most evaluations, ROUGE scores are linear to SO ratios as expected. However, VO has high variance across algorithms and aspects. This is mainly because the semantic volume assumption maximizes the semantic diversity, but sacrifices other aspects like importance by choosing the outlier sentences over the convex hull.

Social posts and news articles are biased to the position aspect while the other two aspects appear less relevant. (Figure 2 (a)) However, XSum requires all aspects equally but with relatively less relevant to any of aspects than the other news corpora.

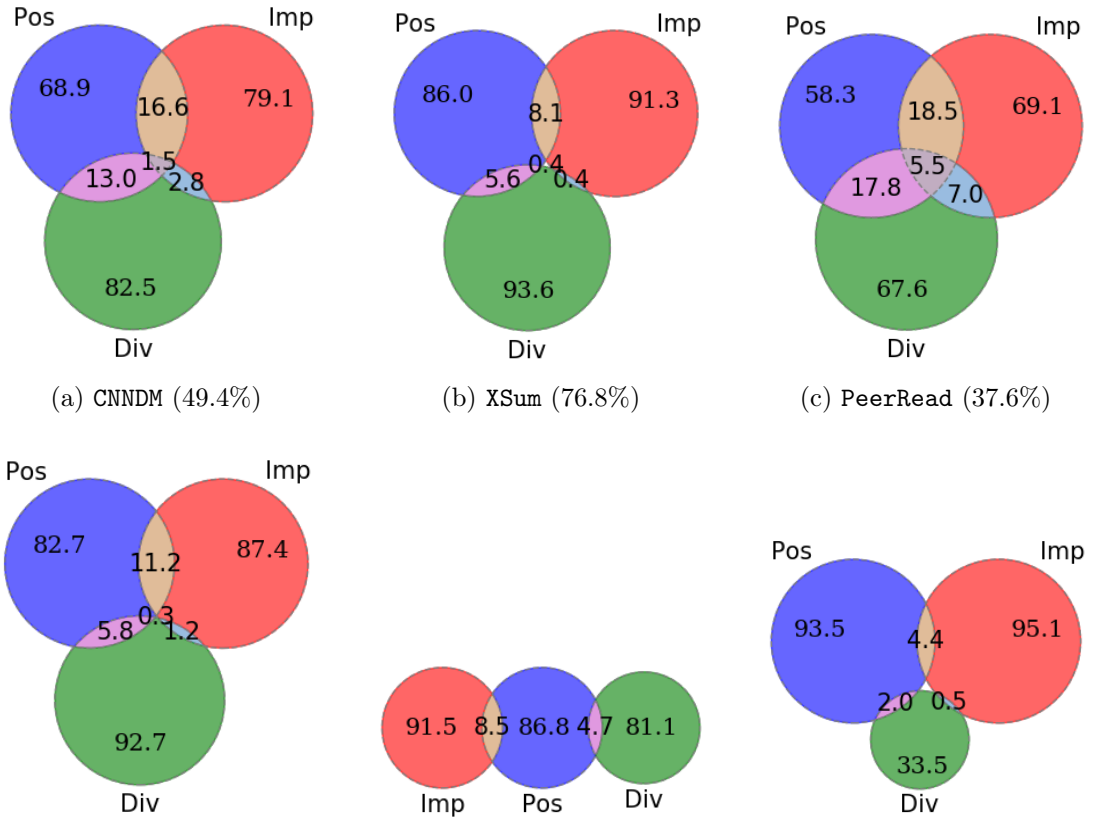
Paper summarization is a well-balanced task. The variance of SO across the three aspects in PeerRead and PubMed is relatively smaller than other corpora. This indicates that abstract summary of the input paper requires the three aspects at the same time. PeerRead has relatively higher SO than PubMed because it only summarizes text in Introduction section, while PubMed summarize whole paper text, which is much difficult (almost random performance).

Conversation, movie script and book summarization are very challenging. Conversation of spoken meeting minutes includes a lot of witty replies repeatedly (e.g., 'okay.' , 'mm -hmm.' , 'yeah.'), causing importance and diversity measures to suffer. MScript and BookSum which include very long input document seem to be extremely difficult task, showing almost random performance.

2.6.2 Intersection between the sub-aspects

Averaged ratios across the sub-aspects do not capture how the actual summaries overlap with each other. Figure 4 shows Venn diagrams of how sets of summary sentences chosen by different sub-aspects are overlapped each other on average.

XSum, BookSum, and AMI have high Oracle Recall. If we develop a mixture model of the three aspects, the Oracle Recall means its upper bound, meaning that another sub-aspect should be considered regardless of the mixture model. This indicates that existing procedures



(a) `CNNDM` (49.4%) (b) `XSum` (76.8%) (c) `PeerRead` (37.6%)

(d) `Reddit` (68.1%) (e) `AMI` (94.1%) (f) `BookSum` (87.1%)

Figure 4: Intersection of averaged summary sentence overlaps across the sub-aspects. We use `First` for **Position**, `ConvexFall` for **Diversity**, and `N-Nearest` for **Importance**. The number in the parenthesis called *Oracle Recall* is the averaged ratio of how many the oracle sentences are **NOT** chosen by union set of the three sub-aspect algorithms. Other corpora are in Appendix B with their Oracle Recalls: `Newsroom`(54.4%), `PubMed` (64.0%) and `MScript` (99.1%).

are not enough to cover the Oracle sentences. For example, `AMI` and `BookSum` have a lot of repeated noisy sentences, some of which could likely be removed without a significant loss of pertinent information.

Importance and Diversity are less overlapped with each other. This means that important sentences are not always diverse sentences, indicating that they should be

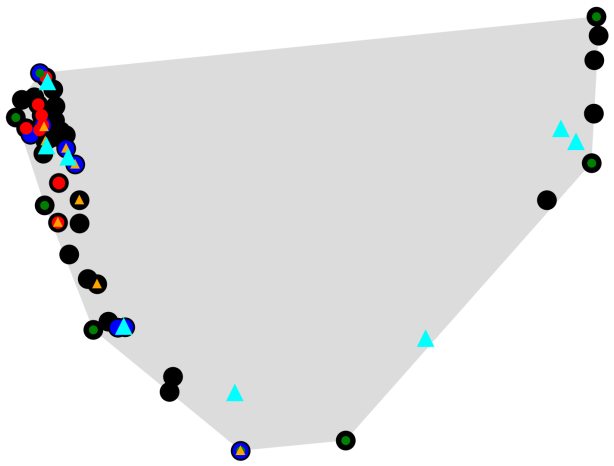
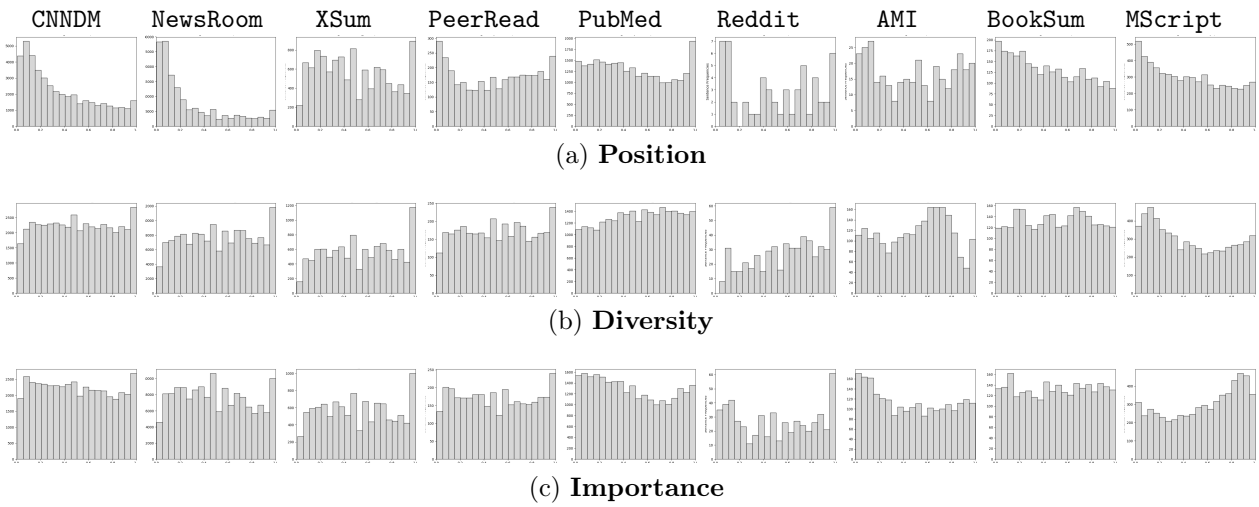


Figure 5: PCA projection of extractive summaries chosen by multiple aspects of algorithms (`CNNDM`). Source and target sentences are black circles and cyan triangles, respectively. The blue, green, red circles are summary sentences chosen by `First`, `ConvexFall`, `NN`, respectively. The yellow triangles are the oracle sentences. Shaded polygon represents a ConvexHull volume of sample source document. Best viewed in color. More examples are in Appendix D

considered together.

2.6.3 Summaries in an embedding space

Figure 5 shows two dimensional PCA projections of a document in `CNNDM` on the embedding space. **Source sentences are clustered on the convexhull border, not in the middle.** . We conjecture that sentences are not uniformly distributed in the embedding space but their positions gradually move over the convexhull. Target summaries reflect different sub-aspects according to the sample and corpora. For example, many target sentences in `CNNDM` are near by `First-k` sentences.



(a) **Position**

(b) **Diversity**

(c) **Importance**

Figure 6: Sentence overlap proportion of each sub-aspect (row) with the oracle summary across corpora (column). y-axis is the frequency of overlapped sentences with the oracle summary. X-axis is the normalized RANK of individual sentences in the input document where size of bin is 0.05. E.g., the first / the most diverse / the most important sentence is in the first bin. If earlier bars are frequent, the aspect is positively relevant to the corpus.

2.6.4 Single-aspect analysis

We calculate the frequency of source sentences overlapped with the oracle summary where the source sentences are ranked differently according to the algorithm of each aspect (See Figure 6). Heavily skewed histograms indicate that oracle sentences are positively (right-skewed) or negatively (left-skewed) related to the sub-aspect.

In most cases, some oracle sentences are overlapped to the first part of the source sentences. Even though their degrees are different, oracle summaries from many corpora (i.e, `CNNDM`, `NewsRoom`, `PeerRead`, `BookSum`, `MScript`) are highly related to the **position**. Compared to the other corpora, `PubMed` and `AMI` contain more top-ranked important sentences in their oracle summaries. News articles and papers tend to find oracle sentences without **diversity** (i.e., right-skewed), meaning that non-diverse sentences are frequently selected as part of the oracle.

	R(O,T)	O∩T		T\S	
		Unigram	Bigram	Unigram	Bigram
CNNDM	42.8	66.0	36.4	14.7	5.7
Newsroom	48.1	60.7	43.4	7.8	3.4
XSum	19.6	30.4	6.9	8.4	1.2
PeerRead	46.3	48.5	27.2	20.1	8.8
PubMed	47.0	52.1	27.7	16.7	6.7
Reddit	30.0	41.0	16.4	13.8	3.8
AMI	32.0	28.1	8.5	10.6	1.5
BookSum	38.9	25.6	8.9	6.7	1.7
MScript	38.9	13.9	4.0	0.3	0.1

Table 3: ROUGE of oracle summaries and averaged N-gram overlap ratios. **O**, **T** and **S** are a set of N-grams (Unigram and Bigram) from **Oracle**, **Target** and **Source** document, respectively. **R(O,T)** is the averaged ROUGE between oracle and target summaries, showing how similar they are. **O∩T** shows N-gram overlap between oracle and target summaries. The higher the more overlapped words in between. T\S is a proportion of N-grams in target summaries not occurred in source document. The lower the more abstractive (i.e., new words) target summaries.

We also measure how many *new* words occur in abstractive target summaries, by comparing overlap between oracle summaries and document sentences (Table 3). One thing to note is that XSum and AMI have less *new* words in their target summaries. On the other hand, paper datasets (i.e., PeerRead and PubMed) include a lot, indicating that abstract text in academic paper is indeed "abstract".

		CNNDM			XSum			PeerRead			PubMed			Reddit			AMI			BookSum			MScript		
		R	SO	R(P/D/I)	R	SO	R(P/D/I)	R	SO	R(P/D/I)	R	SO	R(P/D/I)	R	SO	R(P/D/I)	R	SO	R(P/D/I)	R	SO	R(P/D/I)	R	SO	R(P/D/I)
extractive	**KMeans**	22.2	16.3	14/22/34	9.8	10.0	14/8/90	30.9	28.3	24/28/38	30.6	14.2	31/40/46	14.0	12.5	10/2/82	12.3	2.5	9/6/7	27.2	4.6	5/2/14	9.1	0.3	0/0/9
	MMR	21.6	15.2	12/24/30	9.8	10.0	14/8/**97**	29.6	24.9	26/29/35	30.2	12.9	33/35/42	13.6	11.5	10/3/**88**	12.3	2.5	9/6/7	29.1	6.1	4/0/13	9.5	0.2	0/0/28
	TexRank	19.6	10.3	34/27/27	9.9	8.5	19/11/16	23.9	12.4	32/32/32	18.0	1.7	19/21/20	17.7	16.7	13/9/15	11.1	0.0	17/20/6	6.7	0.0	8/14/8	8.2	0.2	5/9/8
	LexRank	29.3	29.5	71/29/32	11.2	11.9	**61**/15/19	29.0	24.6	**66**/35/38	26.3	7.7	**56**/27/28	18.7	18.8	46/11/19	8.0	0.2	36/21/12	10.5	0.8	20/20/13	12.7	**0.5**	20/9/9
	wILP	23.1	15.6	27/28/29	11.1	2.1	28/19/21	20.2	16.0	23/27/26	15.6	6.0	14/20/18	17.4	13.5	42/16/20	5.1	0.6	17/18/17	4.3	1.3	5/12/7	6.8	0.1	6/8/6
	CL	**31.2**	30.0	**86**/29/31	**11.8**	**14.3**	25/13/19	31.3	21.8	55/35/38	26.3	9.2	41/26/26	19.4	24.0	23/14/23	23.1	10.3	19/23/5	-	-	-/-/-	14.0	0.2	6/8/7
	SumRun	30.5	27.1	68/29/31	11.6	13.1	14/13/19	**34.0**	20.5	38/36/37	29.4	10.8	27/28/27	**20.2**	19.8	23/12/21	**23.8**	11.4	21/23/6	-	-	-/-/-	14.4	0.0	5/9/9
	S2SExt	30.4	28.3	74/28/31	12.0	14.2	17/13/19	33.9	21.1	43/35/37	29.6	10.8	26/28/28	21.5	**34.4**	27/12/26	23.4	11.9	21/24/6	-	-	-/-/-	14.3	0.0	7/9/8
abstractive	**cILP**	27.8	x	43/31/32	10.9	x	49/15/18	28.2	x	35/36/38	27.8	x	23/29/30	17.7	x	**53**/15/17	12.5	x	22/33/10	7.9	x	9/19/12	10.6	x	5/7/7
	S2SAbs	16.3	x	4/4/4	10.4	x	8/7/8	9.9	x	9/9/9	10.2	x	10/10/10	11.9	x	11/7/8	20.3	x	9/12/1	-	-x	-/-/-	14.0	x	6/8/8
	+**Pointer**	23.9	x	20/13/14	15.6	x	12/11/12	13.6	x	13/13/13	11.2	x	11/12/11	14.3	x	14/10/12	23.0	x	11/13/1	-	-x	-/-/-	10.0	x	6/7/7
	+**Teacher**	29.7	x	33/21/22	17.0	x	12/10/12	8.7	x	8/8/8	11.3	x	12/12/11	15.3	x	15/10/11	20.2	x	9/13/1	-	-x	-/-/-	16.0	x	7/10/8
	+**RL**	30.2	x	34/23/24	18.1	x	12/11/12	30.1	x	30/29/28	12.9	x	13/14/13	16.7	x	1/1/14	23.6	x	11/13/2	-	-x	-/-/-	**16.2**	x	7/10/8
ensemble	**asp**(rand)	23.3	19.5	40/**38**/**38**	9.0	9.0	40/**39**/38	29.6	25.5	54/49/52	29.5	13.5	49/**47**/**51**	12.5	5.2	21/11/22	8.9	0.9	44/**50**/**20**	**29.8**	**6.4**	57/33/55	8.4	0.4	**32**/36/37
	asp(topk)	29.1	**30.4**	71/31/31	9.0	8.8	43/**39**/38	30.5	28.2	63/**54**/**57**	29.7	14.0	55/48/52	12.3	15.6	41/**41**/38	9.9	1.5	**99**/24/11	29.6	6.2	**58**/**34**/**56**	8.3	**0.5**	30/**37**/**38**
	ext(rand)	24.2	20.2	39/25/27	10.2	10.9	17/13/23	29.4	23.5	42/37/39	31.7	16.0	37/34/38	14.2	17.7	22/12/13	18.7	5.1	21/28/8	28.6	5.4	37/24/42	6.7	0.0	5/9/13
	ext(topk)	29.4	30.3	58/25/28	11.0	11.8	18/10/37	33.0	**33.0**	54/39/44	**34.1**	**20.5**	41/35/40	16.4	20.8	21/11/52	**23.8**	**13.4**	23/27/6	28.5	5.2	37/24/43	7.4	0.0	6/8/11

Table 4: Comparison of different systems using the averaged ROUGE scores (1/2/L) with target summaries (**R**) and averaged oracle overlap ratios (SO, only for extractive systems). We calculate **R** between systems and selected summary sentences from each sub-aspect (**R(P/D/I)**) where each aspect uses the best algorithm: First, ConvexFall and NNearest. **R(P/D/I)** is rounded by the decimal point. - indicates the system has too few samples to train the neural systems. x indicates SO is not applicable because abstractive systems have no sentence indices. The best score for each corpora is shown in bold with different colors.

2.7 Analysis on System Bias

We study how current summarization systems are biased with respect to three sub-aspects. In addition, we show that a simple ensemble of systems shows comparable performance to the single-aspect systems.

Existing systems. We compare various extractive and abstractive systems: For extractive systems, we use *K-Means* [Lin and Bilmes, 2010], Maximal Marginal Relevance (*MMR*) [Carbonell and Goldstein, 1998], *cILP* [Gillick and Favre, 2009, Boudin et al., 2015], *TexRank* [Mihalcea and Tarau, 2004], *LexRank* [Erkan and Radev, 2004] and three recent neural systems; *CL* [Cheng and Lapata, 2016], *SumRun* [Nallapati et al., 2017], and *S2SExt* [Kedzie

et al., 2018]. For abstractive systems, we use *WordILP* Banerjee et al. [2015] and four neural systems; *S2SAbs* [Rush et al., 2015], *Pointer* [See et al., 2017], *Teacher* [Bengio et al., 2015], and *RL* [Paulus et al., 2017]. The detailed description and experimental setup for each algorithm are in Appendix A.

Proposed ensemble systems. Motivated by the sub-aspect theory [Lin and Bilmes, 2012, 2011], we combine different types of systems together from two different pools of extractive systems: **asp** from the three best algorithm from each aspect and **ext** from all extractive systems. For each combination, we choose the sumary sentences randomly among the union set of the predicted sentences (rand) or the most frequent unique sentences (topk).

Results. Table 4 shows a comparison of existing and proposed summarization systems on the set of corpora in §2.5 except for `Newsroom`[12]. Neural extractive systems such as *CL*, *SumRun* and *S2SExt* outperform the others in general. *LexRank* is highly biased toward the position aspect. On the other hand, *MMR* is extremely biased to the importance aspect on `XSum` and `Reddit`. Interestingly, neural extractive systems are somewhat balanced compared to the others. Ensemble systems seem to have the three sub-aspects in balance, compared to the neural extractive systems. They also outperform the others (either `ROUGE` or `SO`) on five out of eight datasets.

2.8 Conclusion

In this chapter, we first define three sub-aspects of text summarization: position, diversity, and importance. We analyze how different domains of summarization dataset are biased to these aspects. We observe that news articles strongly reflect the position aspect, while the others do not. In addition, we investigate how current summarization systems reflect these three sub-aspects in balance. Each type of approach has its own bias, while neural systems rarely do. Simple ensembling of the systems shows more balanced and comparable performance than single ones.

Our bias study provides meaningful observations for future summarization researches,

[12]We exclude it because of its similar behavior as `CNNDM`.

especially when collecting a dataset and developing a new system. We summarize actionable messages for future summarization research:

- Different domains of datasets except for news articles pose new challenges to the appropriate design of summarization systems. For example, summarization of conversations (e.g., `AMI`) or dialogues (`MSCript`) need to filter out repeated, rhetorical utterances. Book summarization (e.g., `BookSum`) is very challenging due to its extremely large document size. Here current neural encoders suffer from computation limits.
- Summarization systems to be developed should clearly state their computational limits as well as effectiveness in each aspect and in each corpus domain. A good summarization system should reflect different kinds of the sub-aspects harmoniously, regardless of corpus bias. Developing such bias-free or robust models can be very important for future directions.
- Nobody has clearly defined the deeper nature of meaning abstraction yet. A more theoretical study of summarization, and the various aspects, is required. A recent notable example is Peyrard [2019a]'s attempt to theoretically define different quantities of `importance` aspect, and demonstrate the potential of the framework on an existing summarization system. Similar studies can be applied to other aspects and their combinations in various systems and different domains of corpora.
- One can repeat our bias study on evaluation metrics. Peyrard [2019b] showed that widely used evaluation metrics (e.g., ROUGE, Jensen-Shannon divergence) are strongly mismatched in scoring summary results. One can compare different measures (e.g., n-gram recall, sentence overlaps, embedding similarities, word connectedness, centrality, importance reflected by discourse structures), and study bias of each with respect to systems and corpora.

3.0 Calibrating Model Uncertainty in Text Classification

In this chapter, we explore how to improve both classifier's performance and calibration quality by using an auxiliary loss in training. The code for this work is publicly available in https://github.com/THEEJUNG/PosCal/. Note that he following sections are mainly from Jung et al. [2020].

3.1 Introduction

Classification systems, from simple logistic regression to complex neural network, typically predict posterior probabilities over classes and decide the final class with the maximum probability. The model's performance is then evaluated by how accurate the predicted classes are with respect to out-of-sample, ground-truth labels. In some cases, however, the quality of posterior estimates themselves must be carefully considered as such estimates are often interpreted as a measure of confidence in the final prediction. For instance, a well-predicted posterior can help assess the fairness of a recidivism prediction instrument [Chouldechova, 2017] or select the optimal number of labels in a diagnosis code prediction [Kavuluru et al., 2015b].

Guo et al. [2017] showed that a model with high classification accuracy does not guarantee good posterior estimation quality. In order to correct the poorly calibrated posterior probability, existing calibration methods [Zadrozny and Elkan, 2001, Platt et al., 1999, Guo et al., 2017, Kumar et al., 2019] generally rescale the posterior distribution predicted from the classifier after training. Such post-processing calibration methods re-learn an appropriate distribution from a held-out validation set and then apply it to an unseen test set, causing a severe discrepancy in distributions across the data splits. The fixed split of the data sets makes the post-calibration very limited and static with respect to the classifier's performance.

We propose a simple but effective training technique called Posterior Calibrated (**PosCal**) training that optimizes the task objective while calibrating the posterior distribution in

training. Unlike the post-processing calibration methods, **PosCal** directly penalizes the difference between the predicted and the true (empirical) posterior probabilities dynamically over the training steps.

PosCal is not a simple substitute of the post-processing calibration methods. Our experiment shows that **PosCal** can not only reduce the calibration error but also increase the task performance on the classification benchmarks: compared to the baseline MLE (maximum likelihood estimation) training method, **PosCal** achieves 2.5% performance improvements on GLUE [Wang et al., 2018] and 0.5% on xSLUE [Kang and Hovy, 2019], and at the same time 16.1% posterior error reduction on GLUE and 13.2% on xSLUE.

3.2 Related Work

Our work is primarily motivated by previous analyses of posterior calibration on modern neural networks. Guo et al. [2017] pointed out that in some cases, as the classification performance of neural networks improves, its posterior output becomes poorly calibrated. There are a few attempts to investigate the effect of posterior calibration on natural language processing (NLP) tasks: Nguyen and O'Connor [2015] empirically tested how classifiers on NLP tasks (e.g., sequence tagging) are calibrated. For instance, compared to the Naive Bayes classifier, logistic regression outputs well-calibrated posteriors in sentiment classification task. Card and Smith [2018] also mentioned the importance of calibration when generating a training corpus for NLP tasks.

As noted above, numerous post-processing calibration techniques have been developed: traditional *binning* methods [Zadrozny and Elkan, 2001, 2002] set up bins based on the predicted posterior $\hat{p}$, re-calculate calibrated posteriors $\hat{q}$ per each bin on a validation set, and then update every $\hat{p}$ with $\hat{q}$ if $\hat{p}$ falls into the certain bin. On the other hand, *scaling* methods [Platt et al., 1999, Guo et al., 2017, Kull et al., 2019] re-scale the predicted posterior $\hat{p}$ from the softmax layer trained on a validation set. Recently, Kumar et al. [2019] pointed out that such re-scaling methods do not actually produce well-calibrated probabilities as reported since the true posterior probability distribution can not be captured with the often

low number of samples in the validation set[1] To address the issue, the authors proposed a scaling-binning calibrator, but still rely on the validation set.

3.3 Posterior Calibrated Training

In general, most of existing classification models are designed to maximize the likelihood estimates (MLE). Its objective is then to minimize the cross-entropy (Xent) loss between the predicted probability and the true probability over k different classes.

During training time, **PosCal** minimizes the cross-entropy as well as the calibration error as a multi-task setup. While the former is a task-specific objective, the latter is a *statistical objective* to make the model to be statistically well-calibrated from its data distribution. Such data-oriented calibration makes the task-oriented model more reliable in terms of its data distribution. Compared to the prior post-calibration methods with a fixed (and often small) validation set, **PosCal** *dynamically* estimates the required statistics for calibration from the train set during training iterations.

Given a training set $\mathcal{D} = \{(x_1, y_1)..(x_n, y_n)\}$ where x_i is a p-dimensional vector of input features and y_i is a k-dimensional one-hot vector corresponding to its true label (with k classes), our training minimizes the following loss:

$$\mathcal{L}_{\textbf{PosCal}} = \mathcal{L}_{xent} + \lambda \mathcal{L}_{cal} \tag{5}$$

where $\mathcal{L}_{xent}$ is the cross-entropy loss for task objective (i.e., classification) and $\mathcal{L}_{cal}$ is the calibration loss on the cross-validation set. λ is a weighting value for a calibration loss $\mathcal{L}_{cal}$. In practice, the optimal value of λ can be chosen via cross-validation. More details are given in §3.4.

[1] §3.4 shows that the effectiveness of re-calibration decreases when the size of the validation set is small.

Each loss term can be then calculated as follows:

$$\mathcal{L}_{xent} = -\sum_{i=1}^{n}\sum_{j=1}^{k} y_i^{(j)} log(\hat{p}_i^{(j)}) \tag{6}$$

$$\mathcal{L}_{cal} = \sum_{i=1}^{n}\sum_{j=1}^{k} d(\hat{p}_i^{(j)}, q_i^{(j)}) \tag{7}$$

where $\mathcal{L}_{xent}$ is a typical cross-entropy loss with $\hat{p}$ as an updated predicted probability while training. $\mathcal{L}_{cal}$ is our proposed loss for minimizing the calibration loss: q is an true (empirical) probability and d is an function to measure the difference (e.g., mean squared error or Kullback-Leibler divergence) between the updated $\hat{p}$ and true posterior q probabilities. The empirical probability q can be calculated by measuring the ratio of true labels per each bin split by the predicted posterior $\hat{p}$ from each update. We sum up the losses from every class $j \in \{1, 2..k\}$.

We show a detailed training procedure of **PosCal** in Algorithm 1. While training, we update the model parameters (i.e., weight matrices in the classifier) as well as the empirical posterior probabilities by calculating the predicted posterior with the recently updated parameters. For $\mathcal{Q}$, we exactly calculate a label frequency per bin B. Since it is time-consuming to update $\mathcal{Q}$ at every step, we set up the number of $\mathcal{Q}$ updates per each epoch so as to only update $\mathcal{Q}$ at each batch.

3.4 Experiment

We investigate how our end-to-end calibration training produces better calibrated posterior estimates without sacrificing task performance.

3.4.1 Task: NLP classification benchmarks

We test our models on two different benchmarks on NLP classification tasks: GLUE [Wang et al., 2018] and xSLUE [Kang and Hovy, 2019]. GLUE contains different types of general-purpose natural language understanding tasks such as question-answering, sentiment analysis

Algorithm 1 Posterior Calibrated Training

Inputs :

Train set $\mathcal{D}$, Bin B, Number of Classes K

Number of epochs e, Learning rate η

Number of updating empirical probabilities u

Output Θ: Model Parameters

Let $\mathcal{Q}$: Empirical Probability Matrix $\in \mathbb{R}^{B \times K}$

Random initialization of Θ

for $i \in \{1, 2, 3, ...e\}$ **do**

    Break $\mathcal{D}$ into random mini-batches b

    Find a set of steps $\mathcal{S}$ for updating $\mathcal{Q}$ by dividing total number of steps into u equal parts

    for b from $\mathcal{D}$ **do**

        $\Theta \leftarrow \Theta - \eta \nabla_{\Theta} \mathcal{L}_{\textbf{PosCal}} \left(\Theta, \mathcal{Q}\right)$

        if current step $\in \mathcal{S}$ **then**

            $\hat{p} = \text{softmax}(\Theta, \mathcal{D})$

            $\mathcal{Q} \leftarrow CalEmpProb(\hat{p}, B)$

        end if

    end for

end for

and text entailment. Since true labels on the test set are not given from the GLUE benchmark, we use the validation set as the test set, and randomly sample 1% of train set as a validation set. xSLUE [Kang and Hovy, 2019] is yet another classification benchmark but on different types of styles such as a level of humor, formality and even demographics of authors. For the details of each dataset, refer to the original papers.

3.4.2 Metrics

In order to measure the task performance, we use different evaluation metrics for each task. For GLUE tasks, we report `F1` for MRPC, `Matthews` correlation for CoLA, and `accuracy`

for other tasks followed by Wang et al. [2018]. For xSLUE, we use `F1` score.

To measure the calibration error, we follow the metric used in the previous work [Guo et al., 2017]; Expected Calibration Error (`ECE`) by measuring how the predicted posterior probability is different from the empirical posterior probability: $\texttt{ECE} = \frac{1}{K}\sum_{k=1}^{K}\sum_{b=1}^{B}\frac{|B_{kb}|}{n}|q_{kb} - \hat{p_{kb}}|$, where $\hat{p_{kb}}$ is an averaged predicted posterior probability for label k in bin b, q_{kb} is a calculated empirical probability for label k in bin b, B_{kb} is a size of bin b in label k, and n is a total sample size. The lower `ECE`, the better the calibration quality.

3.4.3 Models

We train the classifiers with three different training methods: **MLE**, **L1**, and **PosCal** . **MLE** is a basic maximum likelihood estimation training by minimizing the cross-entropy loss, **L1** is MLE training with L_1 regularizer, and **PosCal** is our proposed training by minimizing $\mathcal{L}_{\textbf{PosCal}}$ (Eq 5). For **PosCal** training, we use Kullback-Leibler divergence to measure $\mathcal{L}_{cal}$. We also report `ECE` with a temperature scaling [Guo et al., 2017] (**tScal**), which is considered the state-of-the-art post-calibration method. For our classifiers, we fine-tuned the pre-trained BERT classifier [Devlin et al., 2019]. Details on the hyper-parameters used are given in Appendix F.

3.4.4 Results

Table 5 and 6 show task performance and calibration error on two benchmarks: GLUE and xSLUE, respectively. In general, **PosCal** outperforms the MLE training and MLE with L_1 regularization in GLUE for both task performance and calibration, though not in xSLUE. Compared to the tScal, **PosCal** shows a stable improvement over different tasks on calibration reduction, while tScal sometimes produces a poorly calibrated result (e.g., CoLA, MRPC).

	Task Perf. (↑)			Calib. ECE (↓)			
Dataset	**MLE**	**L_1**	**PosCal**	**MLE**	**L_1**	**tScal**	**PosCal**
CoLA	56.7	55.3	**58.0**	.242	**.234**	.565	.231
SST-2	92.1	91.4	**92.4**	.144	.155	.143	**.106**
MRPC	88.2	88.2	**88.9**	.228	.229	.400	**.177**
QQP	88.8	88.9	**89.1**	.121	.122	**.054**	.107
MNLI	**84.0**	83.7	83.5	.158	.160	**.080**	.165
MNLI_{mm}	83.7	84.0	**84.2**	.153	.153	**.062**	.149
QNLI	89.9	89.7	**90.0**	.138	.124	.159	.176
RTE	61.7	62.4	**62.8**	.422	.441	**.175**	.394
WNLI	38.0	38.0	**56.9**	.287	.287	.269	**.083**
total	75.9	75.6	**78.4**	.210	.212	.252	**.176**

Table 5: Task performance (left; higher better) and calibration error (right; lower better) on GLUE. We do not include STS-B; a regression task. Note that **tScal** is only applicable for calibration reduction, because the post-calibration does not change the task performance, while **PosCal** can do both.

3.4.5 Analysis

We visually check the statistical effect of **PosCal** with respect to calibration. Figure 7 shows how predicted posterior distribution of **PosCal** is different from **MLE**. We choose two datasets where **PosCal** improves both accuracy and calibration quality compared with the basic MLE: RTE from GLUE and Stanford's politeness dataset from xSLUE. We then draw two different histograms: a histogram of $\hat{p}$ frequencies (top) and a calibration histogram, $\hat{p}$ versus the empirical posterior probability q (bottom). Figure 7(c,d) show that **PosCal** spreads out the extremely predicted posterior probabilities (0 or 1) from MLE to be more well calibrated over different bins. The well-calibrated posteriors also help correct the skewed predictions in Figure 7(a,b).

	Task Perf.(↑)			Calib. ECE(↓)			
Dataset	**MLE**	**L_1**	**PosCal**	**MLE**	**L_1**	**tScal**	**PosCal**
GYAFC	89.1	89.4	**89.5**	.178	.170	.783	**.118**
SPolite	68.7	70.0	**70.9**	.451	.431	**.133**	.238
SHumor	97.4	**97.6**	**97.6**	.050	.047	**.037**	.044
SJoke	**98.4**	98.1	98.3	.032	.037	**.019**	.029
SarcGhosh	42.5	42.5	**42.6**	.912	.912	**.898**	.910
SARC	71.3	71.5	**71.4**	.372	.375	**.079**	.186
SARC_pol	72.7	72.8	**73.8**	.434	.435	**.070**	.383
VUA	80.9	80.8	**81.4**	.268	.276	.687	**.238**
TroFi	76.7	**78.8**	77.4	.278	**.239**	.345	.265
CrowdFlower	22.0	**22.7**	22.6	.404	.413	**.261**	.418
DailyDialog	48.3	47.8	**48.7**	.225	.227	**.117**	.222
HateOffens	93.0	**93.6**	93.5	.064	.059	.100	**.055**
SRomance	99.0	99.0	**100.0**	.020	.020	.023	**.010**
SentiBank	96.7	**97.0**	96.6	.061	.057	**.037**	.054
PASTEL_gender	47.9	**48.1**	47.9	.336	.305	.185	**.143**
PASTEL_age	**23.5**	23.4	22.9	.354	.365	**.222**	.369
PASTEL_country	56.1	56.6	**58.3**	.054	.055	**.019**	.046
PASTEL_politics	46.6	**47.0**	46.8	.394	.379	**.160**	.413
PASTEL_education	24.4	**25.2**	24.7	.314	.332	**.209**	.323
PASTEL_ethnic	**25.3**	24.8	24.8	.245	.243	**.163**	.250
total	64.0	64.3	**64.5**	.272	.269	**.227**	.236

Table 6: Task performance (left; higher better) and calibration error (ECE; lower better) on xSLUE. We do not include EmoBank; a regression task.

To better understand in which case **PosCal** helps correct the wrong predictions from MLE, we analyze how prediction $\hat{p}$ is different between MLE and **PosCal** in test set. Table 7

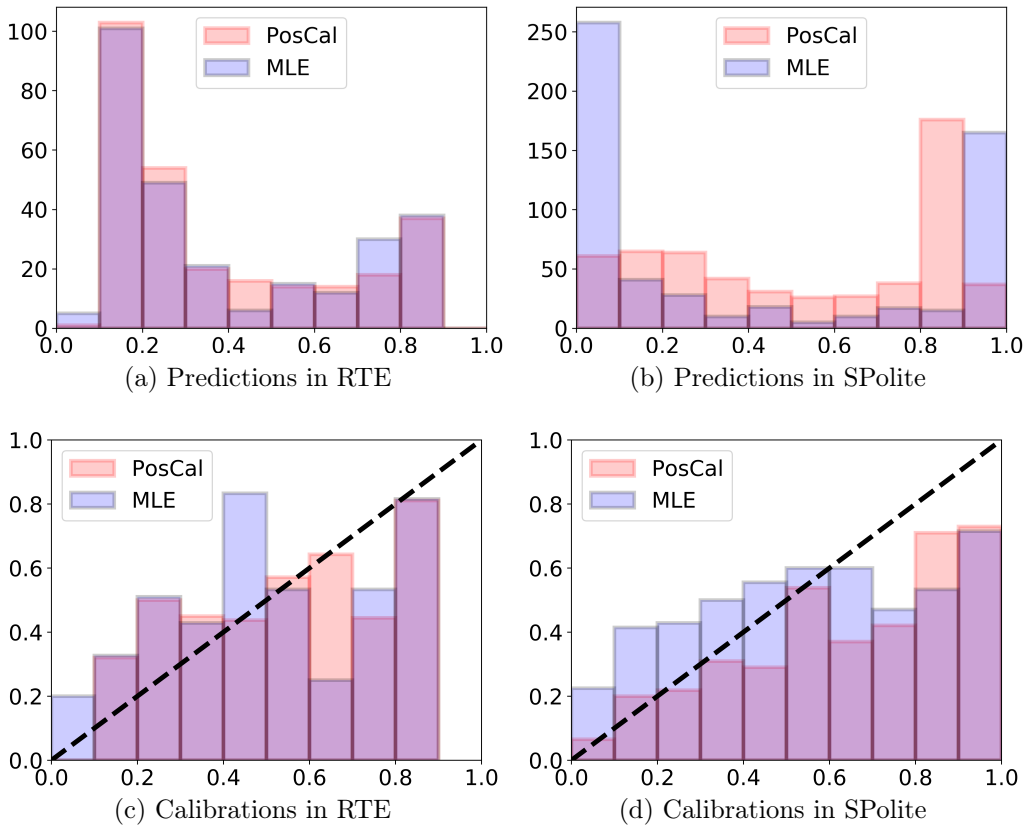


Figure 7: Histogram of predicted probabilities (top) and their calibration histograms (bottom) between **MLE** (blue-shaded) and **PosCal** (red-shaded) on RTE in GLUE and SPoliteness in xSLUE. The overlap is purple-shaded . X-axis is the predicted posterior, and Y-axis is its frequencies (top) and empirical posterior probabilities (bottom). The diagonal, linear line in (c,d) means the expected (or perfectly calibrated) case. We observe that **PosCal** alleviate the posterior probabilities with the small predictions toward the expected calibration. Best viewed in color.

shows the number of correct/incorrect predictions and its corresponding label distributions grouped by the two models. For example, COR by MLE and INCOR by **PosCal** in the fourth row of Table 7 means that there are three test samples that MLE correctly predicts while **PosCal** not.

We find that in most of cases, **PosCal** corrects the wrong predictions from MLE by

Data	**MLE → PosCal** predictions	**Size** (%)	**MLE** avg($\hat{p}$)	**PosCal** avg($\hat{p}$)	**label dist.** 0	1
RTE	COR → COR	164(59.2)	79.2	78.6	42.8	47.2
	COR → INCOR	3(1.1)	59.7	39.0	0	100
	INCOR → COR	9(3.3)	40.6	56.7	100	0
	INCOR → INCOR	101(36.4)	23.6	24.9	27.7	72.3
SPolite.	COR → COR	342(60.3)	95.0	82.6	58.8	41.2
	COR → INCOR	54(9.5)	82.1	26.8	96.3	3.7
	INCOR → COR	60(10.6)	16.9	73.9	15.0	85.0
	INCOR → INCOR	111(19.6)	9.8	21.7	54.0	46.0

Table 7: Size of correct (**COR**) and incorrect (**INCOR**) prediction labels with their averaged $\hat{p}(\%)$ of true labels for **MLE** and **PosCal** on RTE and Stanford's politeness (**SPolite**) dataset. Each has two labels : entail(0) / not entail(1) for RTE, and polite(0) / impolite(1) for SPolite. **PosCal** improves 2.2%/1.1% accuracy than MLE for RTE/SPolite.

re-scaling $\hat{p}$ in a certain direction. In RTE, most inconsistent predictions between MLE and **PosCal** have their posterior predictions near to the decision boundary (i.e., 50% for binary classification) with an averaged predicted probability about 40%. This is mainly because **PosCal** does not change the majority of the predictions but helps correct the controversial predictions near to the decision boundary. **PosCal** improves 3.3% of accuracy but only sacrifices 1.1% by correctly predicting the samples predicted as 'not entailment' by MLE to 'entailment'.

On the other hand, SPolite has more extreme distribution of $\hat{p}$ from MLE than RTE. We find a fair trade-off between two models (-9.5%, +10.6%) but still **PosCal** outperforms MLE.

Table 8 shows examples that only **PosCal** predicts correctly, with corresponding $\hat{p}$ of true label from MLE and **PosCal** (INCOR → COR cases in Table 7). The predicted probability

Data	Sentence	True label	MLE $\hat{p}$	PosCal $\hat{p}$
RTE	**(S1)** Researchers at the Harvard School of Public Health say that people who drink coffee may be doing a lot more than keeping themselves awake - this kind of consumption apparently also can help reduce the risk of diseases. **(S2)** Coffee drinking has health benefits.	entail	49.7	51.3 INCOR → COR
	(S1) The biggest newspaper in Norway, Verdens Gang, prints a letter to the editor written by Joe Harrington and myself. **(S2)** Verdens Gang is a Norwegian newspaper.	entail	43.9	61.9 INCOR → COR
SPolite.	Not at all clear what you want to do. What is the full expected output?	impolite	10.5	74.9 INCOR → COR
	Are you sure that it isn't due to the error that the compiler is thrown off, and generating multiple errors due to that one error? Could you give some example of this?	polite	6.9	57.9 INCOR → COR

Table 8: Predicted $\hat{p}(\%)$ of true label from **MLE** and **PosCal** with corresponding sentences in RTE and SPolite dataset. True label is either entail or not entail for RTE, and polite or impolite for SPolite. Provided examples are the cases only **PosCal** predicts correctly, which correspond to INCOR → COR in table 7.

$\hat{p}$ should be greater than 50% if models predict the true label.

In the first example of RTE dataset, two expressions from S1 and S2 (e.g, "reduce the risk of disease" in S1 and "health benefits" in S2) make MLE confusing to predict, so $\hat{p}$ of true label becomes slightly less than the borderline probability (e.g., $\hat{p} = 49.7\% < 50\%$), making incorrect prediction. Another example of RTE shows how the MLE fails to predict the true label since the model cannot learn the connection between the location of newspaper (e.g., "Norway") and its name (e.g., "Verden Gang"). In the two cases from SPolite dataset, the level of politeness indicated on phrases (e.g., "Not at all" in the first case and "Could you" in the second case) is not captured well by MLE, so the model predicts the incorrect label.

From our manual investigation above, we find that statistical knowledge about posterior probability helps correct $\hat{p}$ while training **PosCal** , so making $\hat{p}$ switch its prediction. For further analysis, we provide more examples in Appendix G.

3.5 Conclusion

In this chapter, we propose a simple yet effective training technique called **PosCal** for better posterior calibration. Our experiments empirically show that **PosCal** can improve both the performance of classifiers and the quality of predicted posterior output compared to MLE-based classifiers. The theoretical underpinnings of our **PosCal** idea are not explored in detail here, but developing formal statistical support for these ideas constitutes interesting future work. Currently, we fix the bin size at 10 and then estimate q by calculating accuracy of p per bin. Estimating q with adaptive binning can be a potential alternative for the fixed binning.

4.0 Quantifying Uncertainty of Individual Observations in Classification

We expect a probabilistic model should be calibrated; that is, predicted probabilities from the model need to be well aligned with their relative frequencies. From the most of studies, a model calibration is “examined" by simple statistics (e.g., expected calibration error), or even by an interval from a rough estimation of relative frequencies. In this paper, we switch a framework of a model calibration in terms of a single point estimation. Now, our main question is how to “interpret" a relative frequency of a single point prediction given. We especially focus on two important issues in this framework; first, how to estimate less biased relative frequencies and second, how to build a robust confidence interval on it. We borrow classical procedures that require weaker assumptions, nearest neighbor estimate and subsampling interval. From the simulations of synthetic and real data settings, we empirically show that (1) nearest neighbor is less biased than existing methods for relative frequency estimation, and (2) subsamling based confidence interval shows tighter bound than the other baselines across various settings. Our framework can be widely used for any classification problem. In particular for high-stakes decision making like a recidivism or a medical diagnosis, such a **customized** calibration confidence interval can provide a better understanding of a connection between predicted probability and its empirical observation.

4.1 Introduction

As statistical and machine learning models become more ubiquitous, humans are increasingly often tasked with making high-stakes decisions based on model outputs. In the context of binary decision making, they are generally understood as predicted probabilities that an event, such as a outbreak of disease [Kavuluru et al., 2015a] or a recidivism [Chouldechova, 2017], occurs. It is natural decision-makers trust that relative frequencies [1] match predicted

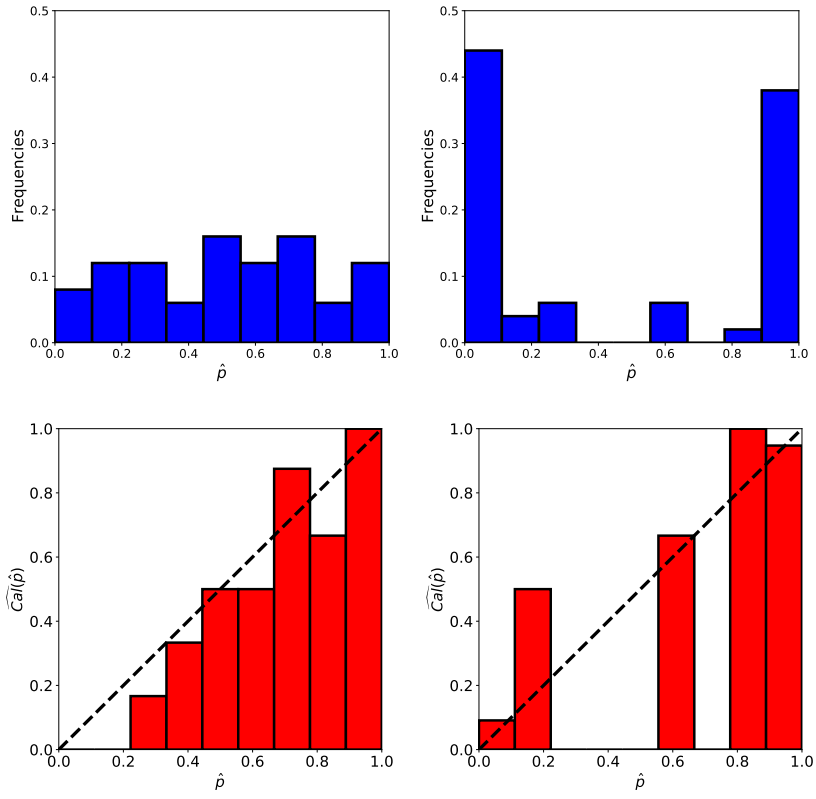
[1] Here, a relative frequency indicates how often a event occurs within the total number of observations under the given predicted probability.

probabilities. For example, in recidivism predictions if a model predicts the probability that a criminal defendant will re-offend at a future point in time as 0.6, then eventually 60% of criminal defendants who assigned probability 0.6 should commit a crime in the future.

Measuring a validity of predicted probabilities was widely studied in meteorology, particularly as concerns weather forecasting. The concept was differently called as a validity [Miller, 1962] or a reliability [Murphy, 1973]. A term 'calibration' was used in the same sense [DeGroot and Fienberg, 1983]. Formally, a model is *perfectly-calibrated* if relative frequencies in the long run equal to all possible predicted probabilities paired. Note that here a model calibration cannot be evaluated by common metrics for classification tasks since it differs from a model performance. In other words, well-calibrated model does not guarantee a highly accurate classification, and/or vice versa. For instance, applying a Bayes classifier to a model which assigns 0.51 to all events that happen and 0.49 to all events that do not happen is perfectly discriminated, but poorly-calibrated since their relative frequencies are 1 and 0 respectively.

In general, a model calibration can be measured in two different ways. A reliability plot [DeGroot and Fienberg, 1983, Guo et al., 2017] is a visual representation. In the plot, predicted probabilities are plotted against corresponding observed relative frequencies. Ideally, histogram bars in plot should align along the diagonal if a model is well-calibrated. More convenient way is to get a statistic summary of calibration; Expected Calibration Error [Naeini et al., 2015] - or `ECE`- is an expected difference between predicted probabilities and observed relative frequencies.

For many real-world statistical models with a discrete probability distribution (e.g., weather forecasters only predicting 0%, 10%, ... 100% chances to rain), such relative frequencies can be easily approximated since multiple observations exist in general for each of possible probabilities as we have enough samples. For infinite forecasters that can output any predicted probabilities in $[0, 1]$, however, it is unlikely the case that for every single predicted probabilities, more than one observations are found. In this case, relative frequencies are approximated using observations in a certain interval of predicted probabilities. Therefore, choosing optimal intervals that minimize a bias of relative frequency approximation is crucial to measure a model calibration.



(a) `Logit` (b) `NN`

Figure 8: Frequency histogram of predicted probabilities (top) and Reliability plot (bottom) of logistic regression (8a) and neural network (8b) in our synthetic setup but with smaller test set (m=50). Note that estimated `ECE` of both models are very close, 0.0127 for `Logit` and 0.0123 for `NN`.

In terms of a calibration measure, even `ECE` is a simple and statistically reasonable representation for validating overall quality of a model calibration, one of the major pitfalls is that this type of statistics cannot capture the variability of observations. It is often the case that predictions in certain ranges may be better- or worse- calibrated than others; see reliability plots (bottom) in Figure 8 for a logistic regression (left) and a neural network (right) trained with a synthetic data described in Section 4.6. Here we use the 50 test samples for both plots and `ECE`. `ECE`s for both logistic regression and neural network are around 0.012, indicating that generally two models are similarly calibrated. However, this

result actually does not align well with our general understanding from the plots; while the logistic regression seems better-calibrated across most of ranges, the neural network does not even have relative frequency estimations in some ranges. It is because neural network tends to predict more extreme probabilities than logistic regression by the nature [Guo et al., 2017]; see Histograms (top) in Figure 8 for a logistic regression (left) and a neural network (right). For neural network, most of predictions are located in the extreme probabilities, range of $[0.0, 0.1]$ and $[0.9, 1.0]$, and assign more weight to the better-calibrated bins.

In practice, humans are often tasked with making distinct individual decisions one at a time. In such cases, the question is not whether overall predictions are biased in estimating long run outcome frequencies, but rather, given a particular circumstance (set of features; individual), how likely is the event to occur given the prediction made. Obviously we can not answer that specific question exactly since we do not know the true probability that the event will occur, but we can provide a plausible range of probabilities based on historical data.

Our research aims to provide interval-based estimations of relative frequencies for given predicted probabilities from models. In particular, we first propose the better way to estimate relative frequencies by using non-parametric statistical methods (e.g, nearest neighbor) instead of widely-used binning methods. We then construct confidence intervals for an individual relative frequency with subsampling. The theoretical proof is simple and highly depending on existing works with some weaker assumption; for example, Biau and Devroye [2015] for nearest neighbor algorithm and Politis and Romano [1994] for subsampling. However, it can be of tremendous practical use for human decision makers, especially for the case that probabilities provided by the prediction has more important meaning than predicted labels in decision making. Also, our work can be easily extended to multi classification tasks and applicable with different non-parametric or sampling methods.

4.2 Related Work

4.2.1 Model Calibration

Model calibration has been widely studied in meteorology, especially for verifying weather forecasting on finite forecasters [Cooke, 1906, Brier et al., 1950, Miller, 1962, Murphy, 1973, DeGroot and Fienberg, 1983, Murphy and Winkler, 1984]. Recently, as it turned out that state-of-the-art neural models are poorly calibrated [Guo et al., 2017], it has been re-spotlighted in machine learning and deep learning studies and extended to explore on infinite forecasters. In particular, most recent works propose methods to improve model calibration; by post-hoc rescaling [Platt et al., 1999, Zadrozny and Elkan, 2001, Guo et al., 2017, Ma and Blaschko, 2021] or training with different loss [Mukhoti et al., 2020, Jung et al., 2020]. In this paper, we do not focus on developing recalibration method, but do explore how to interpret relative frequency estimates on individual points.

`ECE` is a widely used metric to measure the overall quality of model calibration. A quality of `ECE` estimates heavily depends on how to approximate relative frequencies. In general, it is estimated with binning based methods by using equally spaced bin [Naeini et al., 2015] or alternatively bins with an equal number of examples [Nguyen and O'Connor, 2015]. Instead of binning method, non parametric estimation such as kernel density estimation was also proposed [Zhang et al., 2020]. Some works have pointed out that such `ECE` methods are sensitive to parameters. For example, Kumar et al. [2019] showed that for equally-spaced binning method, estimated `ECE` increases as a number of bin increases. Nixon et al. [2019] also argued that binning based `ECE` estimates are sensitive to the binning techniques. Arrieta-Ibarra et al. [2022] even pointed out that both binning and kernel density based `ECE` estimates have trade-offs between statistical confidence for the ability and the variation of functions. In this work, we propose to use nearest neighbor method to estimate relative frequencies which is a simpler version of non parametric estimation. Even though nearest neighbor based `ECE` estimates still need to choose an optimal parameter k, we empirically show that it is much less sensitive than other parameters such as bin size for binning based `ECE` estimates. Recently, Roelofs et al. [2022] compared existing `ECE` methods to find the least biased one,

treating nearest neighbor estimates as a “true" calibration error. We instead argue that the nearest neighbor method itself should be considered as the least-biased `ECE` estimates. In this work, we provide a theoretical background and empirically show how it works on various data setups in Section 4.4.

Meanwhile, a model calibration sometimes identifies with a model specification in Statistics. Classical goodness-of-fit testing method by Hosmer and Lemesbow [1980] and its modifications are still actively used. Since such hypothesis testing results in rejecting calibration rather than showing how model is calibrated, and even well-calibrated models are rejected in large samples [Paul et al., 2013] , alternatively constructing confidence bands on calibration curve is proposed for the purpose of overall model assessment. However, a calibration curve in general assumes a monotonically increasing function, thus, it is approximated with a parametric function [Nattino et al., 2014] or an isotonic regression [Yang and Barber, 2019, Dimitriadis et al., 2022] which can be heavily biased on single observations. [2] Unlike previous works, our work rather aims to construct a calibration interval for a single observation, which we call “customized" interval, to provide an evidence of relative frequency estimates using subsampling method. We continue to provide a theoretical background of customized calibration confidence interval and its applications.

4.2.2 Confidence Interval

As we estimate relative frequencies with a nearest neighbor method, our calibration interval basically becomes a confidence interval of k nearest neighbor regression estimates. Several works studied to build a confidence band for k nearest neighbor regression estimates. For example, Bjerve et al. [1985] applied a uniform confidence bound for k nearest neighbor regression estimates; Eubank and Speckman [1993] considered a bias-corrected method.

Bootstrap is a commonly used technique for building confidence interval of non parametric regression estimates [Härdle and Mammen, 1991, Hall, 1992, Neumann and Polzehl, 1998, Hall and Horowitz, 2013]. On the other hand, subsampling is another statistical method for constructing confidence bands on nonparametric regression estimates. Compared to the

[2] For example, it is not always guaranteed that examples with higher predicted probability align to higher relative frequency.

bootstrap procedure, subsampling requires more weaker assumptions [Bickel and Sakov, 2008] and computationally less expensive if a sample size gets larger [Politis, 2021]. In this work, we mainly use a subsampling method to construct calibration confidence interval which is extended from Politis et al. [1999]. We provide a theoretical background of subsampling method and comparison with other baselines (e.g., bootstrap from Hall and Horowitz [2013] and simultaneous calibration band from Dimitriadis et al. [2022]).

4.3 Model Calibration

Suppose we have train and test sets $\mathcal{D}_{train} = \{Z_1, ...Z_n\}$ and $\mathcal{D}_{test} = \{Z_{n+1}, ...Z_{n+m}\}$ containing n and m observations for each, of the form $(\mathbf{X}_i, Y_i)$ where Y_i denotes the response and $\mathbf{X}_i$ denotes a corresponding vector of d covariates $(X_{i1}, ...X_{id})$. Here we assume that the response is binary where $Y_i \in \{0, 1\}$. For most framework, $\mathcal{D}_{train}$ is first used to fit a model (forecaster) $\hat{F}$ that maps from the covariate space to the probability space $p \in [0, 1]$ where $p = P(Y = 1|X)$ is the posterior probability that the response is positive (=1) given a particular set of covariates $\mathbf{X}$. In terms of the classification, the predicted label $\hat{Y} = 1$ if $\hat{F}(\mathbf{X}) > 0.5$ in general.

Definition 1 (Model calibration)**.** *A predicted forecaster $\hat{F}$ is well calibrated if*

$$P(y = 1|\hat{F}(X) = p) \simeq p$$

holds at least approximately across a range of probabilities p.

Here, we define $Cal(p)$ to a **calibration probability** $P(y = 1|\hat{F}(X) = p)$ for given p. Most common way to quantify the calibration of model $\hat{F}(X)$ is measuring an `ECE` by averaging differences between predicted probabilities (p) and their corresponding calibration probabilities $(Cal(p)$, equally relative frequencies)

$$\texttt{ECE}(\hat{F}) = \mathbb{E}_{\mathbf{X}}(Cal(p) - p)^2. \tag{8}$$

In general, $\texttt{ECE}(\hat{F})$ is estimated on $\mathcal{D}_{test}$. Suppose $\hat{P}_{test} = \{\hat{p}_{n+1}, ...\hat{p}_{n+m}\}$ is the set of predicted probabilities from $\hat{F}$ using the covariates $\mathbf{X}_{test} = (\mathbf{X}_{n+1}, ...\mathbf{X}_{n+m})$ from $\mathcal{D}_{test}$. Then, $\texttt{ECE}(\hat{F})$ can be estimated as

$$\widehat{\texttt{ECE}}(\hat{F}) = \frac{1}{m}\sum_{i=1}^{m}(Cal(\hat{p}_{n+i}) - \hat{p}_{n+i})^2. \tag{9}$$

Here, $Cal(\hat{p}_{n+i})$ needs to be approximated since we cannot get the true calibration probability. Finally, $\texttt{ECE}(\hat{F})$ can be estimated as

$$\widehat{\texttt{ECE}}(\hat{F}) = \frac{1}{m}\sum_{i=1}^{m}(\widehat{Cal}(\hat{p}_{n+i}) - \hat{p}_{n+i})^2. \tag{10}$$

where $\widehat{Cal}(\hat{p})$ is an estimated calibration probability given $\hat{p}$. In general, $\widehat{Cal}(\hat{p})$ cannot be approximated by simply calculating relative frequencies conditioned on a single $\hat{p}$ since mostly $\hat{p}$ is a continuous random variable in $[0, 1]$ except the case of finite forecaster [3]. Thus, we use empirical approximations such that

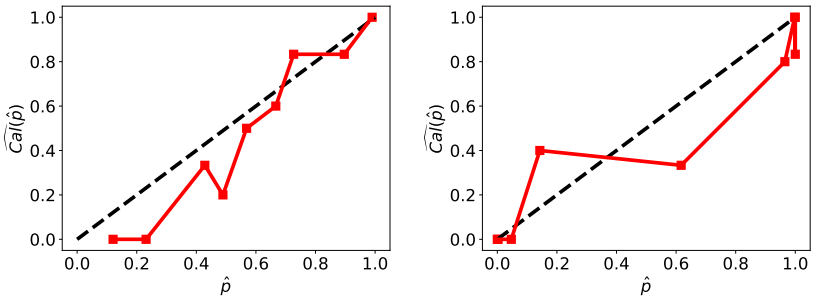
$$\widehat{Cal}(\hat{p}) = \sum_{i=1}^{m}\frac{\mathbf{1}(Y_{n+i} = 1)\mathbf{1}(\hat{p}_{n+i} \in \tilde{B}_{\hat{p}})}{\|\tilde{B}_{\hat{p}}\|} \tag{11}$$

where $\tilde{B}_{\hat{p}} = \{\hat{p}_i : \hat{p}_i \in \hat{P}_{test} \wedge (\hat{p}_i \in N_{\hat{p}})\}$ and $N_{\hat{p}}$ is an interval or bin, including $\hat{p}$ such that $[\hat{p} - I_{lb}, \hat{p} + I_{ub}] \subset [0, 1]$, assuming that the points in $N_{\hat{p}}$ are close enough to $\hat{p}$ for estimating $Cal(\hat{p})$.

4.4 Calibration Probability Estimation

In equation 11, how to implement $N_{\hat{p}}$ is not that much explored so far. Most of previous works use equally-based bins $N_{\hat{p}} = [\frac{b}{B}, \frac{b+1}{B}]$ such that $\hat{p} \in N_{\hat{p}}$ where B is a positive integer and $b = \{0, 1, \ldots, B\}$. It is simple and intuitive, but performs poorly if there is no sufficient number of observations for each bin. In order to use a fixed-bin scheme, we mainly assume

[3] We will see more details about how $Cal(\hat{p})$ can be estimated for a finite forecaster in Section 4.5.1.



(a) `Logit` (b) `NN`

Figure 9: Reliability plot of logistic regression (9a) and neural network (9b) with `ECE` estimators using adaptive binning. For both models, we use the same test output and the bin size as Figure 8.

a CDF from model should be uniformly distributed [Gopal, 2021]. As we have already seen in Figure 8, however, it is not the case for many of statistical machine learning models. Alternative way is using uniform-mass bins, called adaptive bins [Nguyen and O'Connor, 2015]. Instead of partitioning [0,1] with equal length, it assigns an equal number of $\hat{p}_i$, so that size of $N_{\hat{p}}$ should he equal. However, an adaptive binning method can easily make a very wide range of bins if $\hat{p}$ is from highly-skewed distributions. Figure 9 shows a reliability plot of adaptive binning method for two models in Figure 8. Most of bins in a neural network are located in the two extreme and have a very wide bins in the middle (range in [0.1,0.9]), while bins in a logistic regression seem to be evenly distributed in the range. In this case, such wide bins in the neural network cannot be appropriate estimates for $Cal(\hat{p})$.

Note that under binning methods, $\widehat{Cal}(\hat{p})$ *actually* means an averaged $\widehat{Cal}(\hat{p})$ since it remains the same for $\hat{p}$s from the same bin. More precise approach is to build a customized intervals for every $\hat{p}$ observations. A simple way is to build nearest neighbor intervals that for every observation in $N_{\hat{p}}$ no more than $k-1$ other forecasts lie closer to $\hat{p}$. Simply, $\widehat{Cal}(\hat{p})$ is a nearest neighbor regression function estimate. A main advantage of nearest neighbor algorithm is such estimates are consistent under the weak conditions:

Theorem 1 (Universal weak pointwise consistency of $\widehat{Cal}(\hat{p})$)**.** *Let* $l \geq 1$*. Assume that*

$k \to \infty$ *and* $\frac{k}{m} \to 0$. *Then nearest neighbor regression estimate* $\widehat{Cal}(\hat{p})$ *satisfies*

$$\mathbb{E}(|\widehat{Cal}(\hat{p}) - Cal(\hat{p})|^l) \to 0 \text{ at } \mu\text{-almost all } \hat{p} \in [0,1]$$

In particular, $\widehat{Cal}(\hat{p})$ *is universally weakly consistent at* μ*-almost all* $\hat{p}$*, that is,*

$$\widehat{Cal}(\hat{p}) \to Cal(\hat{p}) \text{ in probability at } \mu\text{-almost all } \hat{p} \in [0,1]$$

In other words, for given $\hat{p}$, as we have infinitely many nearest points in $\hat{P}_{test}$ and the size of k diverges much slower than the size of $\mathcal{D}_{test}$, $\widehat{Cal}(\hat{p})$ converges to the true calibration probabilities. Note that theorem above is a simple notation change from Biau and Devroye [2015]; see Theorem 11.1 and Corollary 11.1 in Biau and Devroye [2015] for the detailed proof.

4.5 Customized Calibration Confidence Interval

4.5.1 Finite Forecaster

To begin, suppose that our forecaster $\hat{F}$ can output only a small and finite collection of probabilities. For example, finite forecaster $\hat{F}$ can only output every 10% of probabilities like $\{0.0, 0.1, 0.2, ..., 0.9, 1.0\}$. Actually, this is often the case in common, low-stakes settings such as a weather forecasting. In probability of precipitation (PoP), for example, a weather forecasting service might prefer simply forecasting a 70% chance of rain instead of providing a detailed prediction like 73.6% since simple prediction numbers are more intuitive.

We first demonstrate how we can form the simple point estimate $Cal(\hat{p})$ for finite forecasters. The key simplification that this setting allows for is given a large test dataset, it is increasingly likely that many observations will receive the same forecast $\hat{p}$ from $\hat{F}$. Formally saying, let $\mathbb{P} = \{p_1, p_2, ..., p_k\}$ be a probability outcome space of a finite forecaster $\hat{F}$ and $\mathbf{m} = (m_1, m_2, ..., m_k)$ is a sequence of observations such that $\hat{p} = p_i$. Then,

$\mathbf{m} \sim \mathbf{Multinomial}(m; q_1, q_2, ..., q_k)$ where $q_i > 0$ is a probability that $\hat{F}(\mathbf{X}') = p_i$ and as $m \to \infty$, every sequence in $\mathbf{m} \to \infty$ as well. Thus, for finite forecaster, $N_{\hat{p}} = [\hat{p} - I_{lb}, \hat{p} + I_{ub}]$, both I_{lb} and $I_{ub} \to 0$ as $m \to \infty$.

Here, given a particular covariate $\mathbf{X}'$ for which $\hat{F}(\mathbf{X}') = \hat{p}$, we can form an estimate $Cal(\hat{p})$ by first selecting all test observations predicted exactly as $\hat{p}$ and averaging their corresponding binary response values to get an estimate of the true probability $Cal(\hat{p})$.

For a confidence interval, we describe a simple theoretical background for $Cal(\hat{p})$. Our response Y is a binary variable, assumed i.i.d. in general. It is obvious that $Y \sim \mathbf{Bernouli}(\theta)$ where $\theta \in [0, 1]$ is unknown. We first fit a model $\hat{F}$ with $\mathcal{D}_{train}$, and calculate $Cal(\hat{p}) = P(Y = 1|\hat{F}(\mathbf{X}) = \hat{p})$ with $\mathcal{D}_{test}$ which is independent of $\mathcal{D}_{train}$. Then our test response $\mathbf{Y}_{test} = \{Y_{n+1}, ...Y_{n+m}\}$ can be partitioned into $\mathbf{Y}_{\hat{p}} = \{Y_i : \hat{F}(X_i) = \hat{p}\}$ and $Cal(\hat{p})$ is a parameter of distinct Bernouli trials $\mathbf{Y}_{\hat{p}}$ for any $\hat{p}$ from the finite collection of probabilities. Thus, forming a confidence interval can then be immediately done by applying any interval estimations for the binomial proportion with their required conditions. That is, either parametric or non-parametric confidence interval methods can be used. For the pair comparison with infinite forecasters, we use a subsampling confidence interval procedure that is further discussed in 4.5.2. A summary of this procedure is given in Algorithm 2.

4.5.2 Infinite Forecaster

While simple, the key issue with the previous setup is that most modern statistical and machine learning models do not, at least by default, output predicted probabilities from only a small finite collection. Rather, the forecaster outputs are continuous, taking on any value in $[0, 1]$. This means that given a particular point of interest $\mathbf{X}'$ and forecast $\hat{F}(\mathbf{X}') = \hat{p}$, we cannot simply utilize a subset of observations with the same forecast because (depending on how the forecaster is constructed), there may simply be no other forecasts of exactly $\hat{p}$. In this setting, we cannot replicate the procedure in Algorithm 2 exactly, but we can approximate it by taking subsamples of a nearest neighbor set. In particular, for a user-specified choice of k, we can define $\mathcal{D}_{test}(\hat{p}, k) = \{Z_i | Z_i \in \mathcal{D}_{test} \wedge (\hat{F}(\mathbf{X}_i) \in kNN(\hat{p}'))\}$ where $kNN(\hat{p}')$ is a nearest neighbor based interval $I_{\hat{p}'}$ in Section 4.4 . Then, again, $\widehat{Cal}(\hat{p})$ can be computed by

Algorithm 2 $(1-\alpha)\times 100\%$ Calibration Confidence Intervals for Finite Forecaster

Input:

Train set $\mathcal{D}_{train}$, Test set $\mathcal{D}_{test}$ with m observations, Desired probability $\hat{p}$,

Number of sub-samples S, sub-sample size $d(<< m)$

Output: $(1-\alpha)\times 100\%$ confidence interval for $Cal(\hat{p})$

1: Train a forecaster $\hat{F}: \mathcal{X} \to \mathcal{P} \in [0,1]$ using $\mathcal{D}_{train}$

2: Define $\mathcal{D}_{test}(\hat{p}) = \{Z_i | Z_i \in \mathcal{D}_{test} \wedge \hat{F}(X_i) = \hat{p}\}$

3: Form $\widehat{Cal}(\hat{p})$ by averaging response values in $\mathcal{D}_{test}(\hat{p})$

4: **for** $s \in \{1, 2, 3, ...S\}$ **do**

5: Take sub-sample $\mathcal{D}_s^*$ by sampling d observations in $\mathcal{D}_{test}$ without replacement

6: Form $\widehat{Cal}(\hat{p})_s^*$ by averaging response values in $\mathcal{D}_s^*(\hat{p})$

7: **end for**

8: Define $c_{m,d}(\alpha)$ as α quantile from $\{\sqrt{d}(\widehat{Cal}(\hat{p}) - \widehat{Cal}(\hat{p})_1^*), ..., \sqrt{d}(\widehat{Cal}(\hat{p}) - \widehat{Cal}(\hat{p})_S^*)\}$

9: Take confidence intervals $[\widehat{Cal}(\hat{p}) - \sqrt{m}^{-1} c_{m,d}(1-\frac{\alpha}{2}), \widehat{Cal}(\hat{p}) - \sqrt{m}^{-1} c_{m,d}(\frac{\alpha}{2})]$

averaging response values in $\mathcal{D}_{test}(\hat{p}, k)$.

Unlike finite forecasters, $\widehat{Cal}(\hat{p})$ in infinite forecasters is a non-parametric estimation. Thus, only non-parametric procedures might be considered to build a confidence interval. Here we use a subsampling confidence interval because compared to the other methods like bootstrapping, its validity is easy to be shown under much weaker conditions:

Theorem 2 (Valid subsampling confidence interval)**.** *For $\hat{P}_{test}$ with size m, let $J_m(x, \hat{p})$ is a cumulative distribution function of $\tau_m(\widehat{Cal}(\hat{p}) - Cal(\hat{p}))$ where $\tau_m = \sqrt{m}$. Let d denote a size of subsamples, D_i^* denote i^{th} subsample, and m_d denote $\binom{m}{d}$. Then, the approximation of $J_m(x, \hat{p})$ with subsampling can be defined by*

$$L_{m,d}(x) = \frac{1}{m_d} \sum_{i=1}^{m_d} 1\{\tau_d(\widehat{Cal}(\hat{p})_i^* - \widehat{Cal}(\hat{p})) \leq x\}$$

where $\widehat{Cal}(\hat{p})_i^*$ is an averaged response values in $\mathcal{D}_i^*(\hat{p}) = \{Z_i | Z_i \in \mathcal{D}_i^* \wedge \hat{F}(X_i) = \hat{p}\}$.

Assume there exists a limiting law $J(\hat{p})$ where $J_m(\hat{p})$ weakly converges as $m \to \infty$. Also,

assume $d \to \infty$ *and* $\frac{d}{m} \to 0$. *Define*

$$c_{m,d}(\alpha) = \inf\{x : L_{m,d}(x) \leq \alpha\}$$

Then, two-sided equal-tailed $(1-\alpha) \times 100\%$ *confidence interval for* $Cal(\hat{p})$ *is*

$$\left[\widehat{Cal}(\hat{p}) - \tau_m^{-1} c_{m,d}(1 - \frac{\alpha}{2}), \widehat{Cal}(\hat{p}) - \tau_m^{-1} c_{m,d}(\frac{\alpha}{2})\right]$$

Furthermore, m_s *can be replaced with* $S << m_d$ *if* $S \to \infty$ *as* $m \to \infty$.

Again, we apply existing theorems which is widely used for subsampling; see Politis and Romano [1994] and Politis et al. [2001] for more details. The conditions for both theorems have in common; both procedures require to choose the nearest neighbor size k and the subsample size d much smaller than the test sample size m as $k, d, m \to \infty$. Intuitively, $k << d$ because in subsample at most $k = d$, and if $k = d$, $\widehat{Cal}(\hat{p})$ becomes constant for all $\hat{p}$. In Section 4.6, we empirically show how to choose the optimal k for the fixed d in practice. Once such a set $\mathcal{D}_{test}(\hat{p}', k)$ is established, the procedure in Algorithm 2 can be carried out using this as a replacement for $\mathcal{D}_{test}(\hat{p}', k)$.

4.6 Simulation

In this Section, we focus on answering to two main questions under the controlled data setup. We first show that a nearest neighbor based based method can estimate less biased calibration probabilities than existing binning methods. We then show how customized calibration confidence intervals can be constructed for different type of forecasters in practice.

4.6.1 Data Generation

To generate a synthetic dataset, we follow the simulation setting from Hastie et al. [2017] except for binary labels $\mathbf{Y}$. Suppose that N (sample size), P (dimension size), and ρ (level of autocorrelations among covariates) are given. Then,

- Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$ is i.i.d from $\mathrm{N}(0, \Sigma)$ where $\Sigma \in \mathbb{R}^{P \times P}$ and $\Sigma_{i,j} = \rho^{|i-j|}$.

- Beta coefficients $\beta \in \mathbb{R}^p$ where $\beta_i = 0.5^i$ are drawn. This is a slight modification of ***beta-type 5*** from Hastie et al. [2017].
- Assume there is a linear relationship between $\mathbf{X}$ and log-odds of the true posterior probability. In other words, $P(y = 1|x) = \frac{1}{e^{-(\beta x+\epsilon)}+1}$ where ϵ is a noise and i.i.d from $\mathrm{N}(0, 1)$.
- Binary label is simply generated by $\mathbf{Y} = \mathbf{1}(\mathbb{P}(y = 1|x) > 0.5)$.

For further simulations, we choose a size of train set $n = 1,000$, a size of test set can vary; for example, $m = \{100, 500, 1000, 5000\}$, dimension size $P = 100$ and level of autocorrelations $\rho = 0.35$. Note that we generate bigger test samples to show theoretical validity of $\widehat{Cal}(\hat{p})$ estimation and subsampling procedure.

4.6.2 Optimal k for Nearest Neighbors

We mainly use a nearest neighbor approach for both calibration probability estimation and its confidence interval. For the nearest neighbor, a number of neighbor, k, is an important parameter to decide its performance. However, how to find an optimal k is not clearly defined; rule of thumb is to just choose $k = \sqrt{N}$ where N is a sample size. A toy example provided in Chen et al. [2018] has a similar data setup, where a single predictor $x \in [0, 1]$ and $x \sim$ uniform$(0, 1)$, as our predictor $\hat{p} \in [0, 1]$ as well. In the example, k can be bounded by

$$k(2k - 1)(k - 1) \leq 6N^2 \leq k(2k + 1)(k + 1) \tag{12}$$

, and thus obviously $k \sim N^{\frac{2}{3}}$ for sufficiently large k and N. For example, if we have a sample size 1000, an approximated optimal k is around 100 . Later, we define $k^* = N^{\frac{2}{3}}$ as an optimal k.

4.6.3 Calibration Probability Estimation

We empirically show that nearest neighbor based $Cal(\hat{p})$ estimator is less biased than two existing approximations; fixed-bin and adaptive-bin methods by comparing the difference between $Cal(\hat{p})$ and $\widehat{Cal}(\hat{p})$ estimated by each method. Here we fix the test sample size 1,000 for experiments.

Forecaster. We need to get true calibration probability $Cal(\hat{p})$ to compare. In practice, however, it is impossible to get it as we do not have a population dataset. Thus, we use a simple assumption that our forecaster

$$F_{true}(X) = \hat{p} = P(Y = 1|X)$$

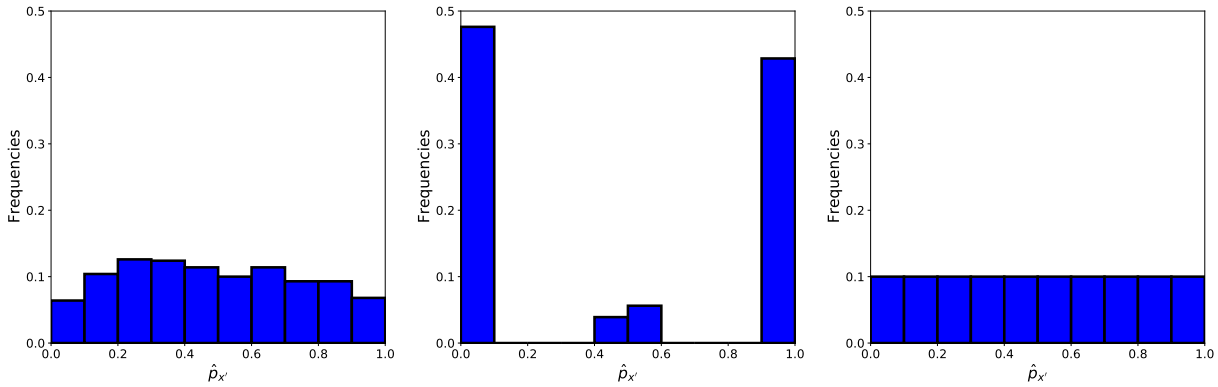
is **perfectly** fitted. In this setup, we even do not need to train model but rather simply use $\hat{p} = P(y = 1|x) = \frac{1}{e^{-\beta x}+1}$ from the synthetic data generation results assuming that we can predict it with F_{true}. Then $Cal(\hat{p}) = \hat{p}$ for all $\hat{p}$ from F_{true} and $\texttt{ECE}(F_{true}) = 0$ as well.

Test Distributions. In order to check how $\widehat{Cal}(\hat{p})$ works on different shape of test distributions, we manipulate the test distribution of $\hat{p}$ as:

- **Norm**: We just keep original distribution shape of $P(Y = 1|X) = \hat{p}$ from the test set. In theory, it just looks like bell-curved centered 0.5 as it follows a logit-normal distribution and the mean and s.t.d of logit(=X) are 0 and 0.35 respectively,.
- **U-shaped**: We choose most of $\hat{p}$ in two extremes, from (0,0.1] and (0.9,1.0] and get a few points in mid, (0.45, 0.55]. This is what we normally get from neural models as it tends to be overconfident on predictions.
- **Uniform**: We simply choose $\hat{p}$ uniformly from (0,1]. This is how current calibration methods are assuming to apply their method (e.g., isotonic regression).

Histograms of $\hat{p}$ with three distributions are in Figure 10.

$\widehat{\texttt{ECE}}(F_{true})$ **Comparison.** In general, $Cal(\hat{p})$ estimator is the least biased if $\widehat{\texttt{ECE}}(F_{true}) > 0$ is the nearest to 0. Here we first compare $\widehat{\texttt{ECE}}(F_{true})$ from $Cal(\hat{p})$ estimators. For each method, parameters determining the size of $N_{\hat{p}}$ should be optimized by minimizing $\widehat{\texttt{ECE}}(F_{true})$; for example, number of bins (B) for fixed and adaptive bin methods and k for nearest neighbor method. We measure $\widehat{\texttt{ECE}}(F_{true})$ across all possible B or k, from 1 to 1000 and report the minimum $\widehat{\texttt{ECE}}(F_{true})$ in Table 9. With optimal k, nearest neighbor method is always less-biased than the others in any types of test distributions. For bin-based methods, optimal bin sizes are between 10 to 20 which aligns to $B = 15$ that is used in previous works by convention. On the other hand, optimal k of nearest neighbor method comes from from 59

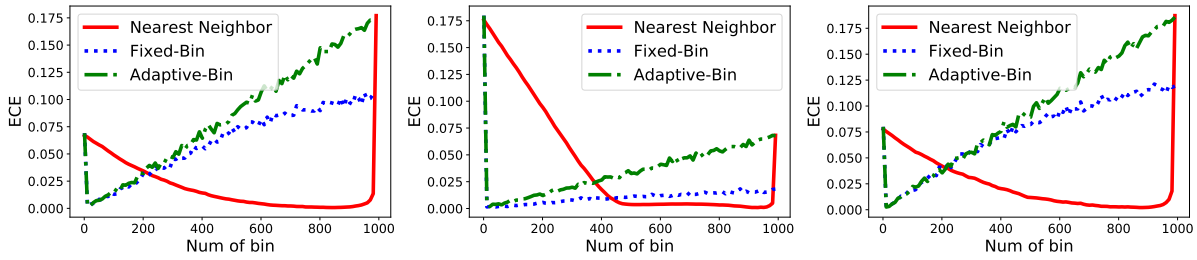


(a) Normal Dist. (b) U-shaped Dist. (c) Uniform Dist.

Figure 10: Histogram of $\hat{p}$ on three different distribution setups for 1,000 test sets.

	Norm Dist.	U-shaped Dist.	Uniform Dist.
Fixed-bin	0.0017 (15)	0.0012 (18)	0.0022 (13)
Adaptive-bin	0.0017 (10)	0.0015 (19)	0.0025 (13)
Nearest Neighbor	**0.0007** (153)	**0.0009** (59)	**0.0021** (115)

Table 9: Minimum $\widehat{\texttt{ECE}}(F_{true})$ for $Cal(\hat{p})$ with corresponding optimal bin (k) sizes. Smaller $\widehat{\texttt{ECE}}(F_{true})$ means less-biased in our setup. For all three test distributions, nearest neighbor method shows significantly less biased in general.



(a) Normal Dist. (b) U-shaped Dist. (c) Uniform Dist.

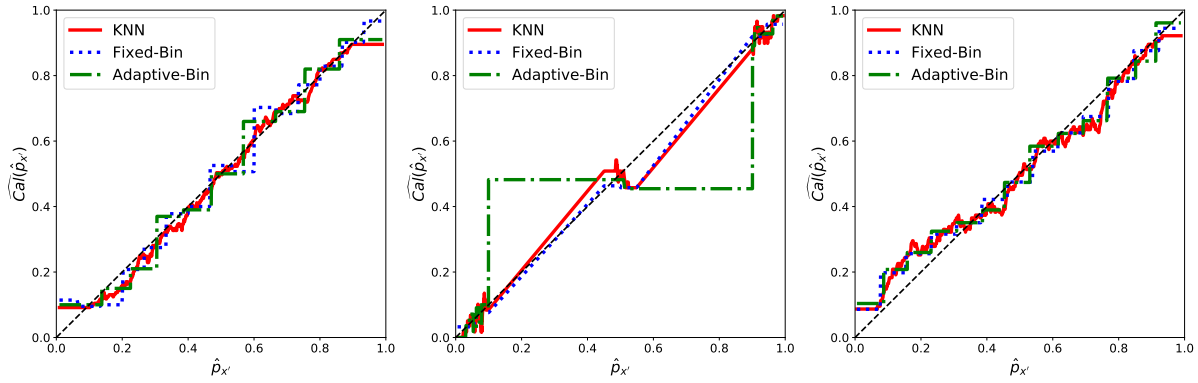
Figure 11: Comparison of $\widehat{\texttt{ECE}}(F_{true})$ across unique number of intervals (e.g., number of bins (B) or N-k) with test size 1,000. Note that for any methods, $\widehat{\texttt{ECE}}(F_{true})$ should be equal in two extreme interval sizes (1 and 1,000).

to 153. Compared to $k^* = 100$, **Uniform** seems to have a close k. We continue to check if $\widehat{\texttt{ECE}}(F_{true})$ of nearest neighbor method is less biased with different k.

We observe $\widehat{\texttt{ECE}}(F_{true})$ across all B or k sizes for three test distributions in Figure 11. In graphs, x-axis corresponds to the unique number of intervals (bins), for example B for binning methods and (N-k) for nearest neighbor method. Nearest neighbor method is in general less-biased in terms of $\widehat{\texttt{ECE}}(F_{true})$ across number of intervals unless k is too big, for example, bigger than around 800 for **Norm** and **Uniform** test distributions. We also mark the range of k where nearest neighbor estimator is always less-biased than the minimum $\widehat{\texttt{ECE}}(F_{true})$ from other methods in Table 10. For instance, with **Norm** test distribution, this

	Norm Dist.	U-shaped Dist.	Uniform Dist.
Fixed-bin	[81,259]	[34,101]	[99,134]
Adaptive-bin	[81,263]	[30,103]	[85,149]

Table 10: Range of Ks where nearest neighbor method has smaller $\widehat{\texttt{ECE}}(F_{true})$ than minimum $\widehat{\texttt{ECE}}(F_{true})$ from the other methods. We find that in general this range contains optimal k 100.



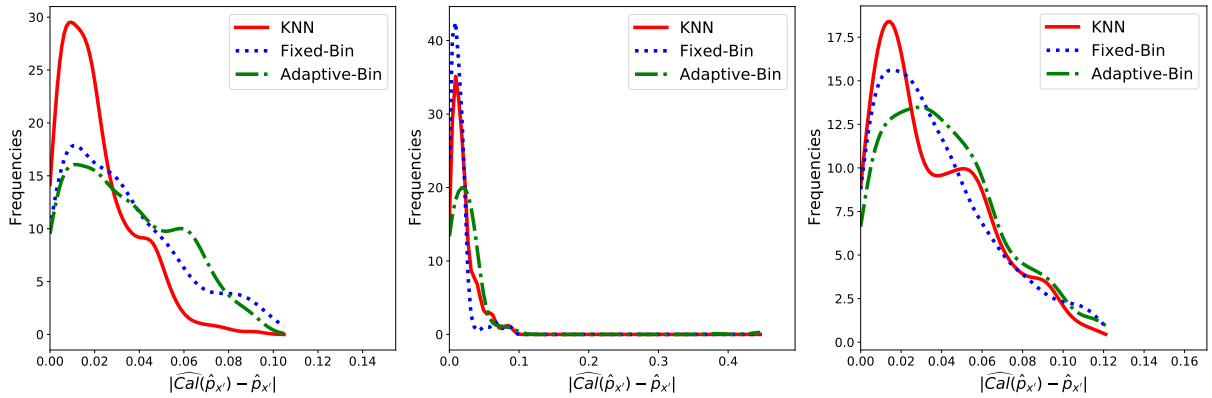
(a) Normal Dist. (b) U-shaped Dist. (c) Uniform Dist. Dist.

Figure 12: $\tilde{p}$ estimators vs $\hat{p}$ across test distributions. As our setup makes $\hat{p}$ equals to true $\tilde{p}$, this should align well to the diagonal line.

ranges are around 80 to 260, which means that even we manually choose any k between 80 to 260, nearest neighbor method is always less biased than the other two. For **U-shaped** and **Uniform** test distribution, the range of k is much narrower. However, it still contains $k^* = 100$, indicating that nearest neighbor method not sensitive to choose an optimal k as long as it is reasonable, and it is robust on different shapes of test distributions.

Pointwise estimators comparison. Instead of checking overall bias from methods using $\widehat{\texttt{ECE}}(F_{true})$, we measure biases of individual $Cal(\hat{p})$ approximations. Graphs of $Cal(\hat{p})$ vs. $\widehat{Cal}(\hat{p})$ across test distributions are in Figure 12. For $Cal(\hat{p})$ approximations, we use an optimal B or k reported in Table 9. Like a reliability plot in Figure 8, the graph should also align to the diagonal line if $\widehat{Cal}(\hat{p})$ is close to $Cal(\hat{p})$. While bin-based methods has stepwise approximations in graphs, nearest neighbor approximations are more customized using moving intervals for each point estimations. In particular, for **Norm** and **U-shaped** test distributions, nearest neighbor method obviously seems to be less biased than the others.

We also compare the frequency of absolute differences between true and estimated calibration probabilities ($|Cal(\hat{p}) - \widehat{Cal}(\hat{p})|$) across test distributions in Figure 13. In this graph, right-skewed graph means pointwise $Cal(\hat{p})$ estimators are closer to true $Cal(\hat{p})$, indicating that the method is less biased. In **Norm** and **Unif** test distributions, nearest neighbor



(a) Normal Dist. (b) U-shaped Dist. (c) Uniform Dist.

Figure 13: Smoothed frequency graph on $|\hat{p} - \tilde{p}|$. Right-skewed graph means that $\tilde{p}$ is close to $\hat{p}$ in pointwise, which means less-biased estimators.

method is more right-skewed than the others. On the other hand, **U-shape** test distribution has more right-skewed graph from fixed-bin method, but has long-tail frequencies between 0.2 to 0.25 that make $\widehat{\texttt{ECE}}(F_{true})$ be greater than nearest neighbor method.

4.6.4 Customized Calibration Confidence Interval

We continue to show how to build a customized confidence interval of $\widehat{Cal}(\hat{p})$ approximations with nearest neighbor method on different forecasters. For baselines, we use simple non-parametric bootstrap confidence interval from Hall and Horowitz [2013] and calibration band proposed by Dimitriadis et al. [2022].

Model Performance and Calibration Error. We first train three different forecasters using $\mathcal{D}_{train}$; k nearest neighbor regression (`KNN`), logistic regression (`Logit`) and neural networks (`NN`). For each forecaster, we use some parameters; for `KNN`, $K = 10$; for `NN`, hidden layers size $H = (1000 \times 512 \times 512)$ and activation function 'ReLU'.

Task performance and calibration error on three different models of different test set size are in Table 4.6.4. In general, `Logit` and `NN` outperforms to `KNN` since simulated datasets are generated by linear relation of $\mathbf{X}$ and $\mathbf{Y}$. In particular, `Logit` outperforms `NN`; this result is

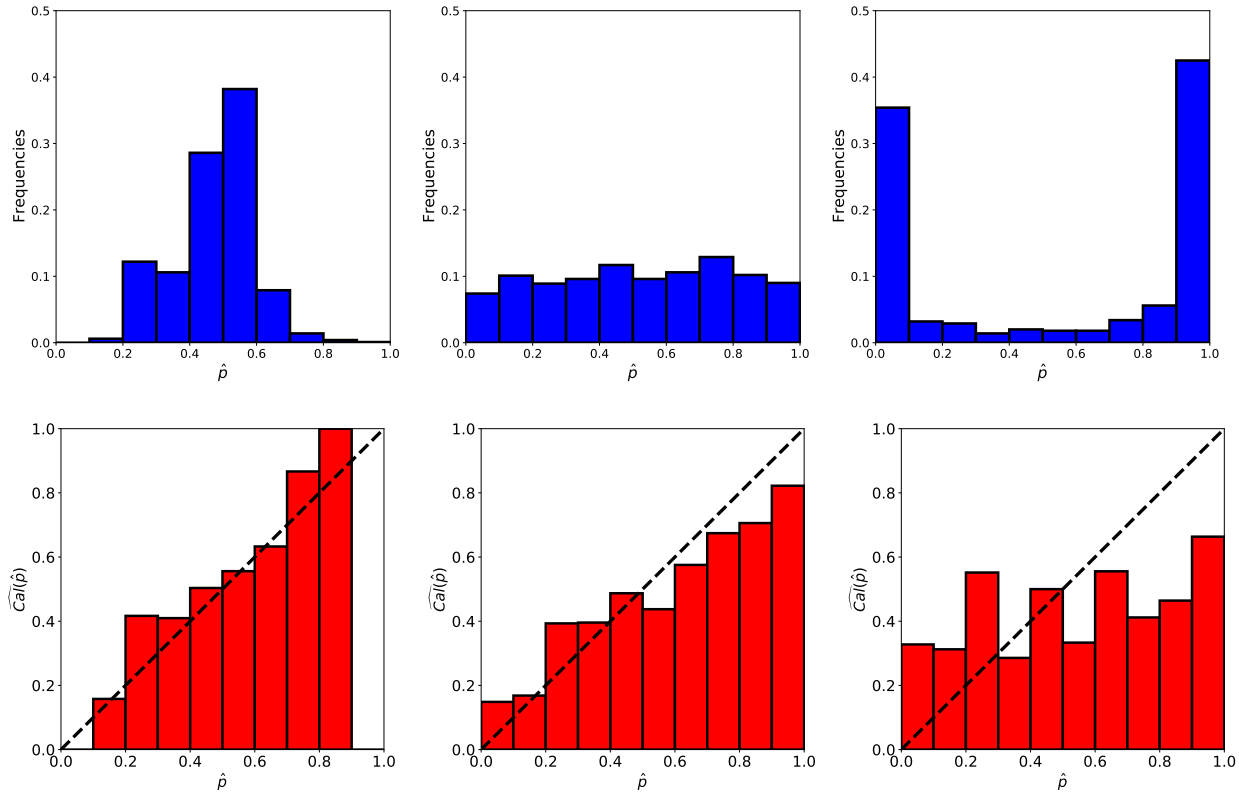
Forecaster	m=100			m=500			m=1,000			m=5,000		
	$\widehat{\texttt{ECE}}_{fix}$	$\widehat{\texttt{ECE}}_{nn}$	Acc.	$\widehat{\texttt{ECE}}_{fix}$	$\widehat{\texttt{ECE}}_{nn}$	Acc.	$\widehat{\texttt{ECE}}_{fix}$	$\widehat{\texttt{ECE}}_{nn}$	Acc.	$\widehat{\texttt{ECE}}_{fix}$	$\widehat{\texttt{ECE}}_{nn}$	Acc.
KNN	**0.0172**	**0.0151**	59.0%	**0.0060**	**0.0069**	55.2%	**0.0034**	**0.0044**	56.9%	**0.0016**	**0.0021**	57.6%
Logit	0.0303	0.0255	**61.0%**	0.0097	0.0094	**68.0%**	0.0086	0.0089	**65.5%**	0.0081	0.0086	**65.7%**
NN	0.1403	0.1255	60.0%	0.0786	0.0718	67.2%	0.0933	0.0909	64.7%	0.0890	0.0896	64.5%

Table 11: Test accuracy and ECE estimations of fixed-bin ($\widehat{\texttt{ECE}}_{fix}$) and nearest neighbor method ($\widehat{\texttt{ECE}}_{nn}$). For fixed-bin ECE estimation, we use bin size (B) 10. For nearest neighbor ECE estimation, we use $k = \sqrt{m}^{\frac{2}{3}}$. For each test size, best scores are **bold**.

not aligned with a general understanding of NN's better performance because a true posterior probability distribution comes from logistic function as well .

In terms of model calibration, NN is actually poor-calibrated than the other two. This result corresponds to the visual representation in Figure 14. In reliability plots, both KNN and Logit are almost perfectly calibrated over all ranges, while NN does not calibrate well, even for two extremes where most of predictions are located in. Histogram plots also show that empirical distribution of $Cal(\hat{p})$ is way much different according to forecasters; while Logit seems uniformly distributed, NN has a u-shaped distribution, highly skewed to both extremes.

Customized Calibration Confidence Interval. Our proposed calibration confidence intervals with different test sample sizes are in Figure 15. As mentioned in Algorithm 2, we use a classic subsampling procedure. We simply set up a subsample size $s = \frac{m}{5}$ and the number of subsamples $S = 1000$, which are small/large enough to fit our weaker conditions from Theorem 2. We then manually choose an arithmetic sequence of $\hat{p} = \{0, \frac{1}{20}, \frac{2}{20}, ..., 1\}$ and construct confidence intervals for individual points. In general, confidence interval gets narrower as test sample size increases across different models. Especially, interval of NN actually contains the conventional border line probability of 0.5 except extremes. This indicates that even though NN shows a great task performance in general, we cannot guarantee that individual prediction points with mid probabilities actually aligns to its actual relative frequencies.

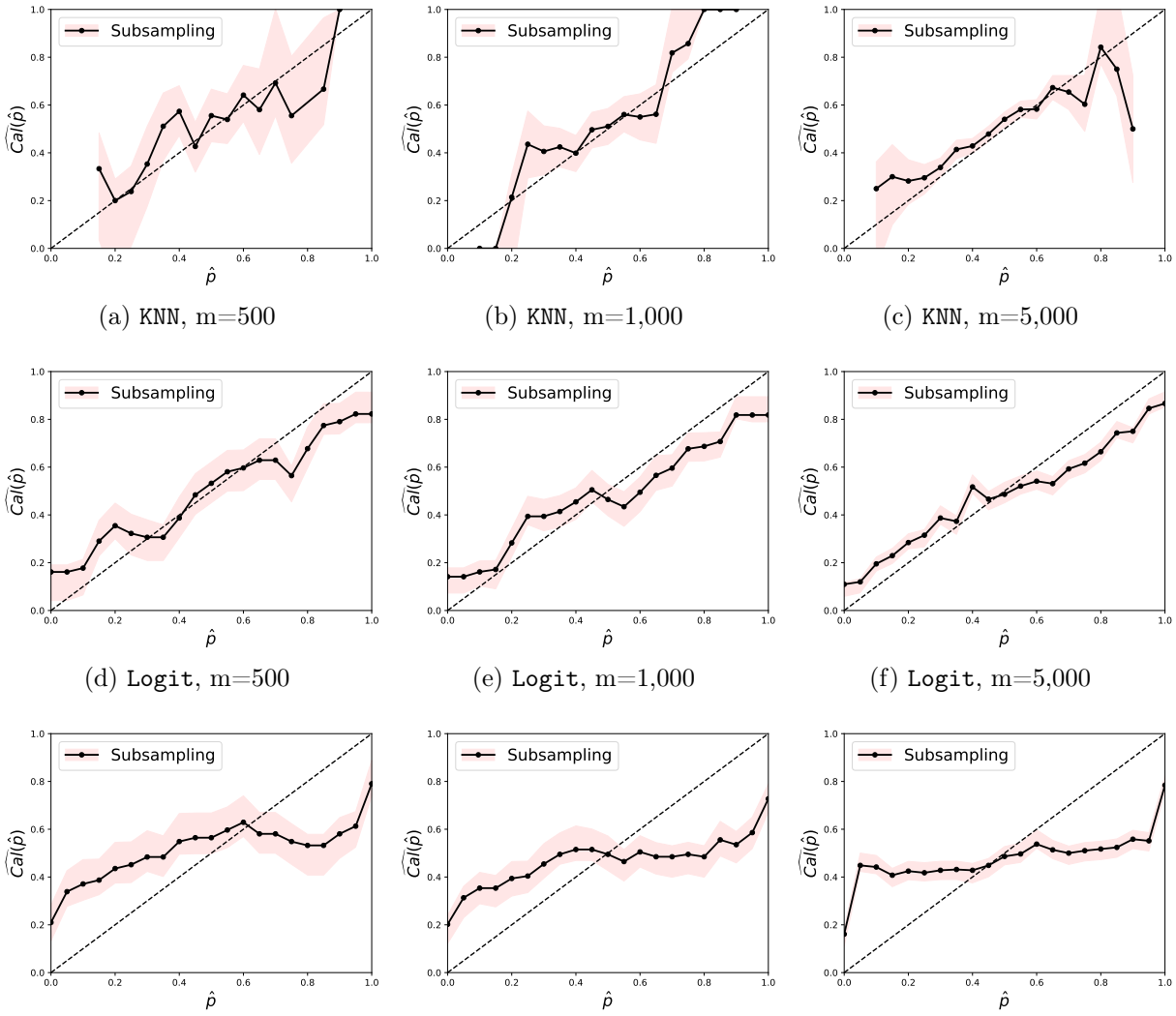


(a) `KNN` (b) `Logit` (c) `NN`

Figure 14: Frequency histogram of $\hat{p}$ (top) and reliability plot (bottom) for `KNN` (14a), `Logit` (14b), and `NN` (14c) in our synthetic data setup with $m = 1,000$.

We further analyze how different subsampling size affect to the interval width in Figure 16. We test subsample sizes $S \in \{\frac{m}{10}, \frac{m}{5}, \frac{m}{2}\}$ where $m = 1,000$. Even though bigger subsample size tends to have tighter confidence bounds, there is no big differences as long as subsample size is small enough.

Comparison with baseline confidence bands. We now compare our proposed calibration confidence interval with two baselines discussed in Section 4.2. Again, we do point out that baseline confidence intervals are conceptually different from our work. For example, most existing works for calibration confidence band [Dimitriadis et al., 2022, Yang and Barber, 2019] target model specification and it does not thoroughly care about the quality of predicted calibration probability $\hat{Cal}(\hat{p})$ by simply drawing it with monotonic regression



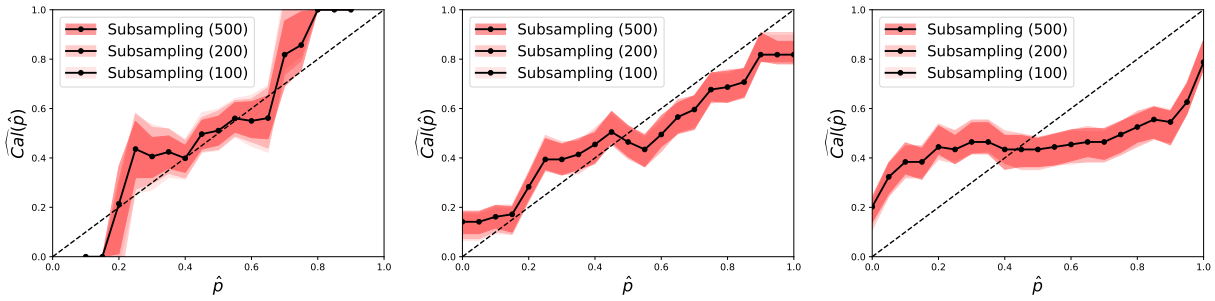
(a) KNN, m=500 (b) KNN, m=1,000 (c) KNN, m=5,000

(d) Logit, m=500 (e) Logit, m=1,000 (f) Logit, m=5,000

(g) NN, m=500 (h) NN, m=1,000 (i) NN, m=5,000

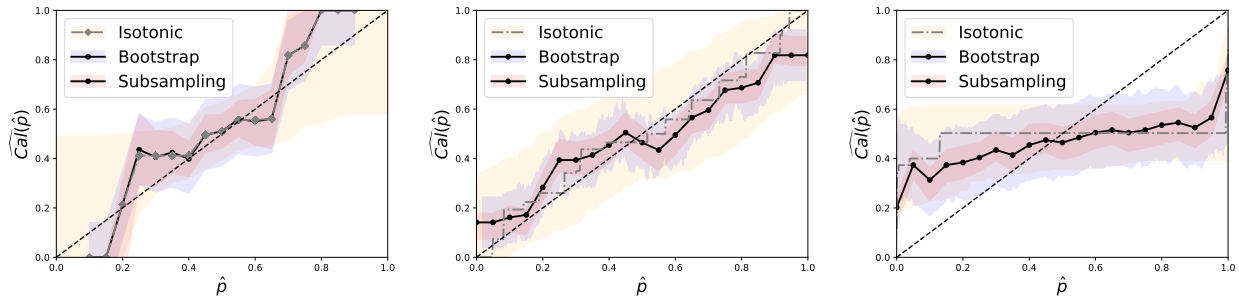
Figure 15: Customized calibration confidence intervals of KNN (top), Logit (center) and NN (bottom) with different test sample size (m) 500 (left), 1000 (middle), and 5000 (right) using subsampling method. For subsampling size, we fix $\frac{m}{5}$ for each.

techniques (e.g., isotonic regression). Here we use the recently proposed method by Dimitriadis et al. [2022] as a baseline. We also customize the general bootstrap approach from Hall and Horowitz [2013] for the calibration confidence interval. See Appendix H for more details about baseline bootstrap method for calibration confidence interval.



(a) KNN (b) Logit (c) NN

Figure 16: Calibration confidence intervals for $m = 1,000$ with different subsampling size. As long as subsampling size is small enough, there exist no big difference of interval width.



(a) KNN (b) Logit (c) NN

Figure 17: Comparison of calibration confidence intervals for m=1,000 on KNN (17a), Logit (17b), and NN (17c).

Figure 17 shows a comparison of calibration confidence intervals across forecasters with test size 1,000. Following Figure 15, we use a subsample size $s = 200$ and the number of subsample $S = 1,000$. We can easily check that subsampling confidence interval has much tighter bounds compared to the other methods for all models. For example, bootstrap confidence interval shows more bumpy bounds overall points and has wider intervals than subsampling method. On the other hand, Dimitriadis et al. [2022]'s interval on isotonic regression estimators show the widest intervals. In terms of point estimations, in addition, such a monotonic function based method cannot accurately approximate the point if its

Forecaster	$\widehat{\texttt{ECE}}_{fix}$	$\widehat{\texttt{ECE}}_{nn}$	Acc.
KNN	0.0022	0.0027	66.08%
Logit	0.0023	0.0028	67.02%
NN	**0.0011**	**0.0015**	**68.13%**

Table 12: Test accuracy and ECE estimations of fixed-bin ($\widehat{\texttt{ECE}}_{fix}$) and nearest neighbor method ($\widehat{\texttt{ECE}}_{nn}$) for COMPAS prediction. For fixed-bin ECE estimation, we use bin size (B) 10. For nearest neighbor ECE estimation, we use $k = m^{\frac{2}{3}}$. Best scores are **bold**.

calibration probability should be lower than the former points. Therefore, for a single point, we cannot guarantee the quality of approximation and even its confidence bound.

4.7 COMPAS application

In this section, we provide an example of a calibration confidence interval application in the real world problem by predicting a recidivism using COMPAS data [4]. Here, we predict if a convicted criminal is likely to re-offend in two years. As Dressel and Farid [2018] showed that simple linear models with fewer variable achieve a same accuracy with COMPAS software trained by full (=137) features, we only use seven features following their work. [5] From 7,214 total defendants, we randomly split 80%/20% of train/test sets. We then train three models again; KNN, Logit, and NN using the same hyper-parameters except that we use 5 hidden layers with size 10 for NN.

Test accuracy and ECE estimations on COMPAS prediction are in Table 12. All three models have comparable task performance and model calibration. In particular, NN shows the best performance and model calibration.

We continue to check a visual representation of model calibration in Figure 18. We show

[4] https://github.com/propublica/compas-analysis/

[5] age, sex, number of juvenile misdemeanors, number of juvenile felonies, number of prior (nonjuvenile)

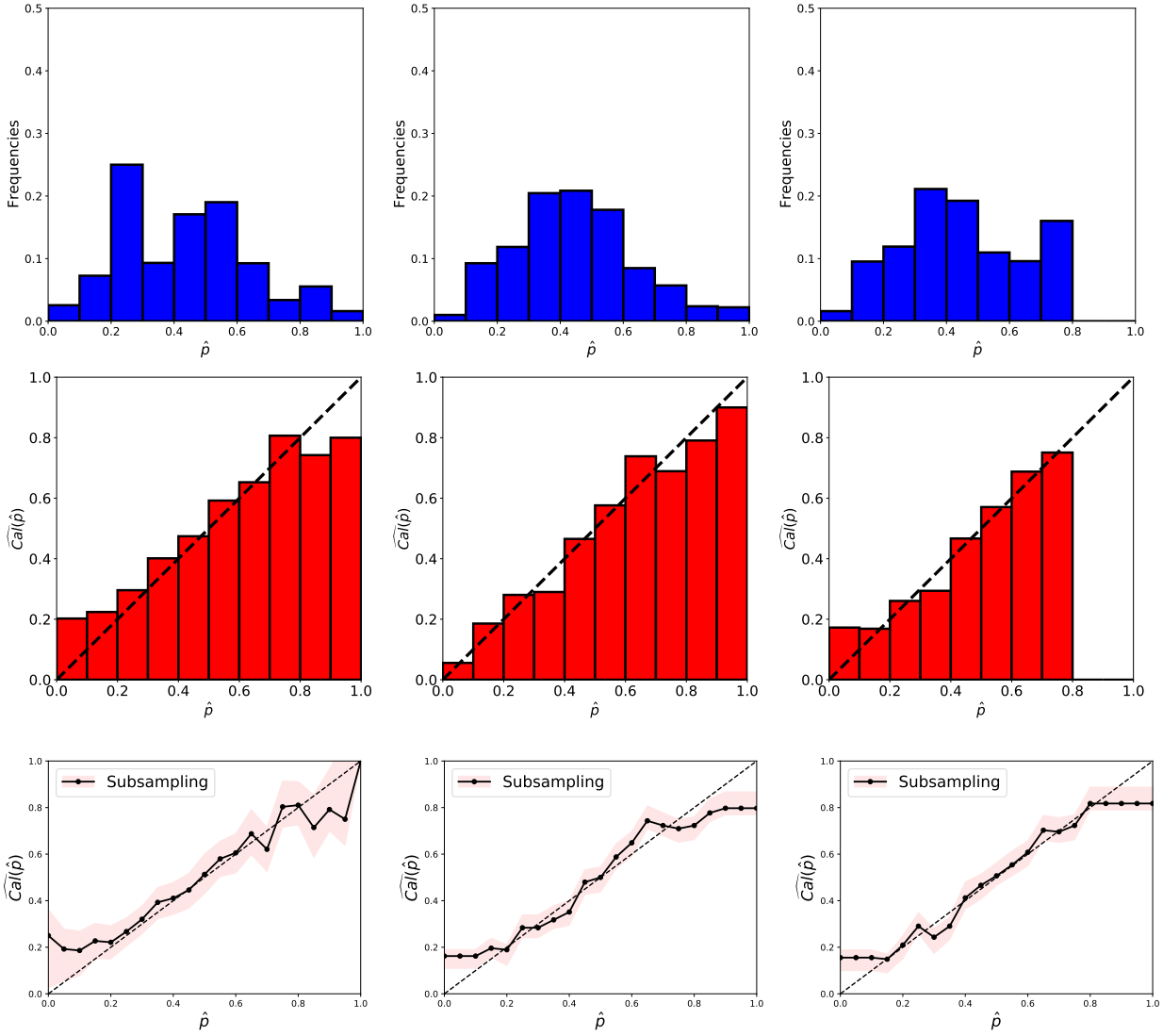
histograms of $\hat{p}$ (top), reliability plots (center), and calibration confidence intervals (bottom) of three models. For some hyper-parameters, we keep them same as the previous sections. We find that even if the models achieve comparable task performances, their frequency histograms of $\hat{p}$ are totally different. For `NN`, none of $\hat{p}$ exceeds 0.8, meaning that the model does not output higher probability on the test set given.

In the calibration confidence intervals, `Logit` and `NN` have much tighter bounds than `KNN`. As `NN` shows the same $Cal(\hat{p})$ approximation over the $\hat{p} > 0.8$, for every point estimation in this range would not have any discrimination in terms of calibration probability; in other words, we may interpret that any of convicted criminals who assign > 0.8 predicted probability from `NN` would have chance to re-offend around 0.8 in the future.

4.8 Conclusion

In this chapter, we propose a new framework to understand a model calibration in terms of a single point estimation. In particular, we borrow the classical procedures such as k nearest neighbor regression and subsampling procedure to estimate point calibration probability and to provide their confidence. From simple simulations on both synthetic and real world setups, we empirically shows the validity of our proposed procedure compared to the baselines.

The big assumption behind the model calibration is that once a model is trained, it is **fixed**; in other words, uncertainty from a model is not generally considered in a model calibration. Exploring a combined framework of a customized model calibration and a model uncertainty can be a great future work. For example, Booth et al. [1992] provided a general framework of bootstrap estimation with conditional distribution that seems appropriate for a combined framework. Exploring other non-parametric methodologies also can be a potential future work; for example, we do not thoroughly explore kernel density estimation for $Cal(\hat{p})$ approximation as it requires more hyper-parameters to be tuned (e.g., bandwidth and kernel function). Finding an optimal non-parametric method based on specific experimental setups might be helpful to apply our ideas.



(a) KNN (b) Logit (c) NN

Figure 18: Comparison of histograms (top) , reliability plots (center), and calibration confidence intervals (bottom) for COMPAS prediction on KNN (18a), Logit (18b), and NN (18c).

5.0 Conclusions

Measuring and understanding the uncertainty behind statistical models should be deeply studied since it helps not only correct the potential bias of models, but also improve the model performance eventually. In this work, we explore three topics to quantify uncertainty, particularly focusing on natural language processing tasks. In detail, we measure data and model biases from the trained model in Chapter 2. In Chapter 3 and 4, we investigate the model calibration in classification by proposing a new framework of its practical use or improving the quality.

In the model calibration, however, we restrict the task as a binary classification since it has a simpler statistical assumption, i.e., target response is binomial distributed. The task extension to the different setups can be an interesting and practical future work. For example, many prior works [Mukhoti et al., 2020, Ni et al., 2019, Widmann et al., 2019] already studied the model calibration in multi classification where a target response can be from multiple candidate classes. Our two ideas can be easily extended to the multi classification task by assuming the target to be multinomial distributed. Further applications on multi classification can be an easy stratch from our study.

More practical but hard task is a multi label classification, that is, having multiple targets from the candidate classes. For example, finding a genre of movies can be a multi label classification as a single movie can belong to multiple genres such as “fantasy", “sf" and “romance". Unlike other classification tasks, model calibration under the multi label classification is rarely explored even if existing state-of-the-art models show the poorly calibrated results. It is because the statistical assumptions behind the multi label classification is much more complicated as candidate classes are not independent. Rather, the classes are highly correlated in general, thus, such correlation should be thoroughly considered under the appropriate setting. Thus, studying for a calibration method and a measurement on multi label classification can be another interesting topic that heavily affects the real world applications.

Appendix A Details on Systems and Setup for text summarzation

For extractive systems, *K-Means* rank sentences clusters by descending order of cluster sizes, and then using a greedy algorithm [Lin and Bilmes, 2010] to select the nearest sentences to the centroid. Maximal Marginal Relevance (*MMR*) finds sentences which are highly relevant to the document but less redundant with sentences already selected for a summary. *cILP* [Gillick and Favre, 2009, Boudin et al., 2015] weights sub-sentences and maximizes their coverage by minimizing redundancy globally using Integer Linear Program (ILP). *TexRank* [Mihalcea and Tarau, 2004] automatically extracts keywords using Levenshtein distance between the text keywords. *LexRank* [Erkan and Radev, 2004] uses module centrality for ranking the keywords. In addition, we also use the recent three neural extractive systems: *CL* [Cheng and Lapata, 2016], *SumRun* [Nallapati et al., 2017], and *S2SExt* [Kedzie et al., 2018], where each has a little variation in their extraction architecture[1].

In training *CL*, *SumRun*, and *S2SExt*, we use upweight positive labels to make them proportional to the negative labels. We use 200 embedding size of GloVe [Pennington et al., 2014] pre-trained embeddings with 0.25 dropout on embeddings, fixing it not to be trained during training. We use CNN encoder with 6 window size as [25, 25, 50, 50, 50, 50] feature maps. We use 1-layer of sequence-to-sequence model with 300 size of LSTM and 100 size of MLP with 0.25 dropout. *SumRun* uses 16 size of segment and 16 size of position embeddings.

For abstractive systems, we use *WordILP* [Banerjee et al., 2015] that produces a word graph of important sentences and then choose sentences from the word graph employing a ILP solver. We also use incremental sequence-to-sequence models: a basic *S2SAbs* [Rush et al., 2015] with *Pointer* network [See et al., 2017], with teacher forcing *Teacher* [Bengio et al., 2015], and with reinforcement learning on the evaluation metrics, and *RL* [Paulus et al., 2017].

In training *S2SAbs*, *Pointer*, *Pointer*, and *RL*, we use 150 hidden size of GRU with 300 size of GloVe embeddings. *Pointer* uses maximum coverage function using NLL loss. *Teacher* uses 0.75 ratio of teach forcing with exponential decaying function. and *RL* uses 0.1 ratio of

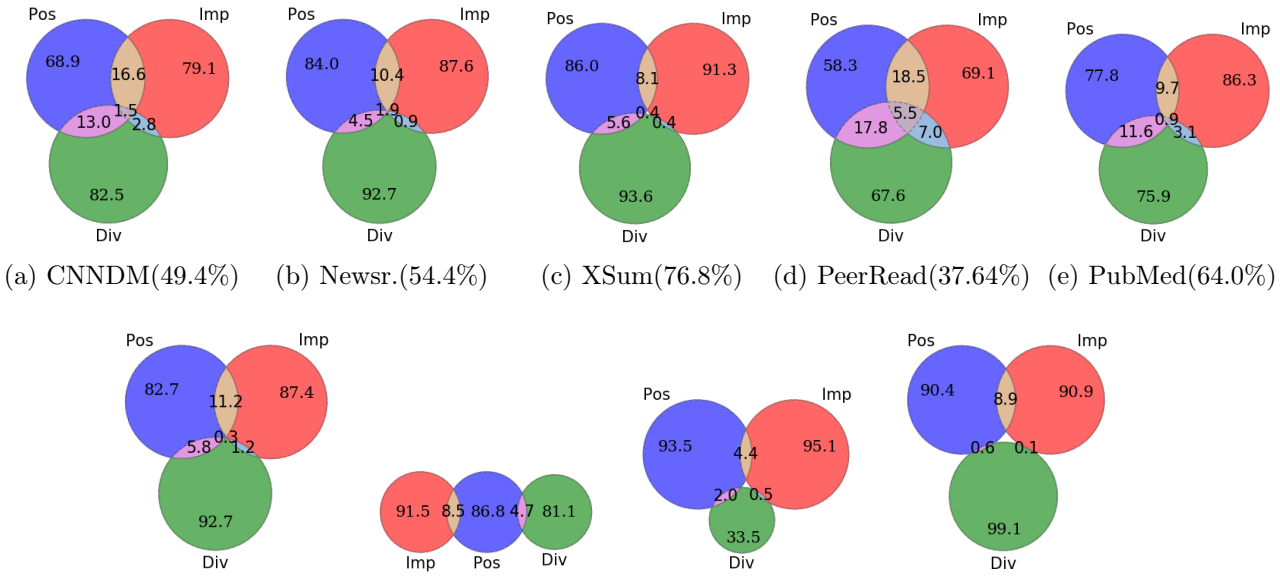
[1]See Kedzie et al. [2018] for a detailed comparison.

RL optimization after the first epoch of *S2SAbs* training. We use 4 size of beam searching at decoding. We use 32 batch size with adam optimizer of 0.001 learning rate.

For `MScript`, the original dataset has no data split, so we randomly split it by 0.9, 0.05, 0.05 for train, valid, test set, respectively.

Appendix B Venn Diagram for All Datasets

Sentence Venn diagrams among three aspects and oracle for all datasets are shown in Figure 19. `Newsroom` has an analogous pattern to `XSum`. Compared to `PeerRead`, , `PubMed` has relatively less sentence overlap between FIRST-K and the other two aspects. `MScript` has extremely small oracle sentence overlaps to all three aspects. However, it is mainly because of the characteristics of the dataset: it has long source documents (1k sentences on average) with short (5 sentences on average) summary.



(a) CNNDM(49.4%) (b) Newsr.(54.4%) (c) XSum(76.8%) (d) PeerRead(37.64%) (e) PubMed(64.0%)

(f) Reddit(68.1%) (g) AMI(94.1%) (h) BookSum(87.1%) (i) MScript (99.1%)

Figure 19: Venndiagram of averaged summary sentence overlaps across the the sub-aspects for all datasets.

Appendix C Full ROUGE F Scores for Corpus Bias Analysis

In Table 13, we provide a full list of ROUGE F scores for all datasets w.r.t three sub-aspects. We find that in `MScript`, the best algorithms for each of ROUGE-1/2/L are different.

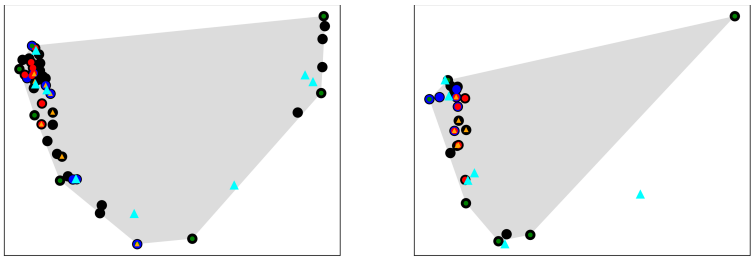
		CNNDM	NewsRoom	XSum	PeerRead	PubMed
		R-1/2/L	R-1/2/L	R-1/2/L	R-1/2/L	R-1/2/L
	Random	26.6/6.7/23.9	15.2/2.8/12.2	14.9/1.8/11.2	38.2/11.1/34.3	41.3/11.3/37.6
	Oracle	51.5/28.5/48.6	53.4/40.2/50.7	27.9/7.5/23.2	56.6/29.5/52.7	58.2/27.9/54.8
Pos.	First-k	**39.1/17.1/35.8**	**36.9/25.9/33.9**	14.8/1.4/11.1	**41.4/16.8/37.9**	37.8/10.2/34.7
Pos.	Last-k	23.5/4.7/21.1	11.5/2.0/9.5	13.2/1.5/10.1	39.1/12.4/35.1	39.1/11.8/35.9
Pos.	Middle-k	29.4/8.6/26.4	17.4/5.3/14.4	14.7/1.7/11.0	40.4/12.5/36.3	39.5/10.8/36.3
Div.	ConvexFall	29.5/8.6/26.6	15.0/4.0/12.7	13.6/1.3/10.5	40.4/12.8/36.3	39.0/10.3/35.3
Div.	Heuristic	29.2/8.7/26.3	14.9/4.1/12.7	13.6/1.3/10.5	39.7/12.4/35.6	38.1/9.8/34.5
Imp.	N-Nearest	29.7/9.3/26.9	18.9/6.1/15.7	**15.7/2.0/11.7**	**41.4**/13.2/37.3	**43.1/12.7/39.5**
Imp.	K-Nearest	30.6/10.5/27.8	19.1/6.8/16.0	15.0/1.8/11	41.0/14.0/36.9	40.0/12.3/36.6

		Reddit	AMI	BookSum	MScript
		R-1/2/L	R-1/2/L	R-1/2/L	R-1/2/L
	Random	17.6/3.7/14.2	17.4/2.2/16.3	41.6/7.0/39.6	12.2/0.7/11.3
	Oracle	38.5/17.8/33.8	42.8/12.3/40.9	52.0/14.7/50.2	33.5/7.3/31.7
Pos.	First-k	**21.8/6.2/17.8**	16.4/2.3/15.5	**40.8/7.6/38.9**	10.3/**1.1**/9.4
Pos.	Last-k	16.4/3.7/13.4	11.1/1.7/10.5	37.6/5.8/36.1	**13.4**/0.9/12.1
Pos.	Middle-k	17.4/3.2/13.8	16.1/1.9/15.2	39.4/6.6/37.7	12.1/0.6/11.2
Div.	ConvexFall	17.3/3.2/14.2	**20.4/2.5/19.1**	24.3/3.9/22.6	12.8/0.7/11.9
Div.	Heuristic	17.2/3.2/14.2	15.7/1.5/15.0	38.2/6.2/36.4	9.7/0.5/9.1
Imp.	N-Nearest	20.6/4.4/16.5	1.9/0.1/1.8	39.3/6.9/37.4	13.1/0.8/**12.2**
Imp.	K-Nearest	15.1/3.6/12.3	0.0/0.0/0.0	30.9/5.0/29.5	1.0/0.0/1.0

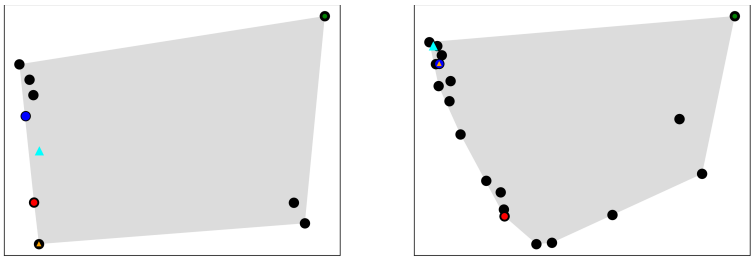
Table 13: Full ROUGE-1/2/L F-Scores for different corpora w.r.t three sub-aspects algorithms.

Appendix D Documents in an Embedding Space: for All Datasets

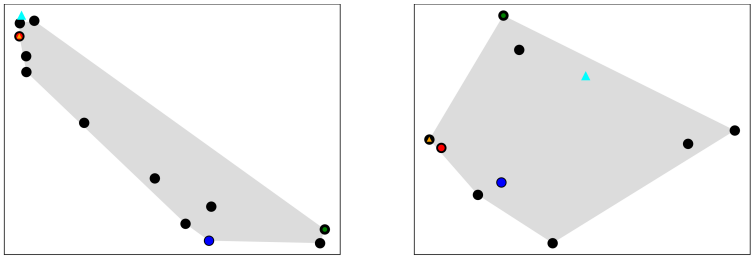
In Figure (20,21), we have more two-dimensional PCA projection examples for source documents from all datasets. We find a weak pattern about where target sentences lie on according to the number of them. For example, from `XSum` and `Reddit` which have a single target sentence, we investigate that some target sentences are located in the middle of `ConvexHull`, which are far from any source sentences.



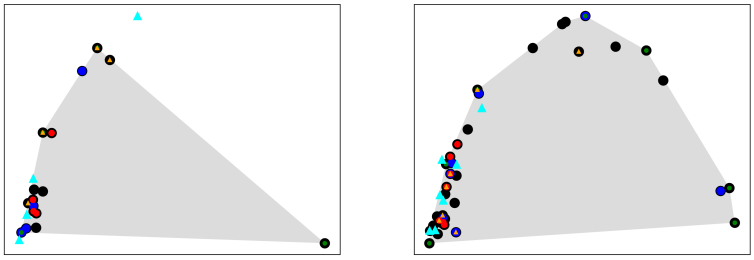
(a) CNNDM



(b) NewsRoom

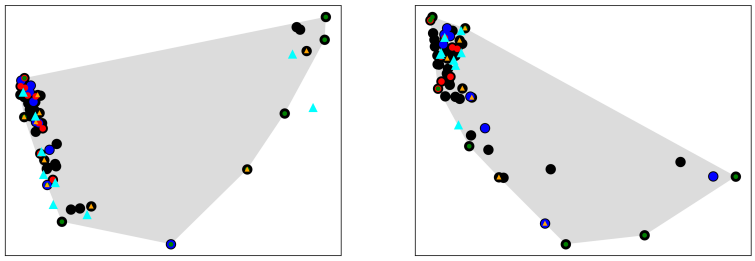


(c) XSum

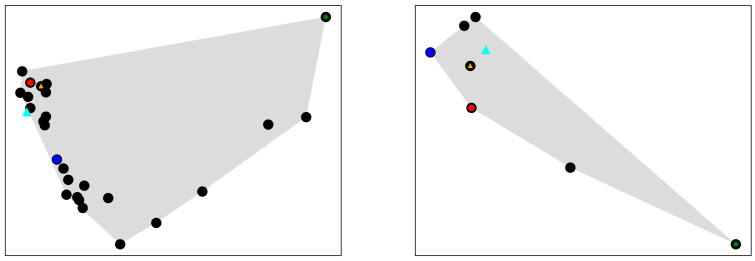


(d) PeerRead

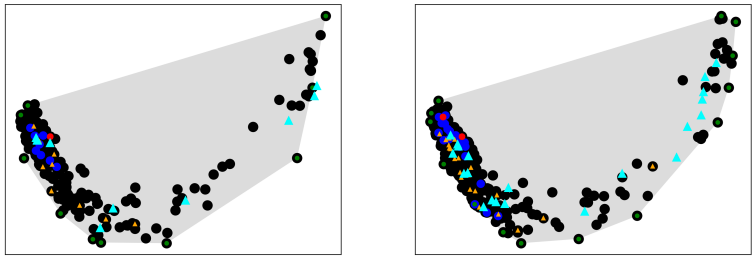
Figure 20: PCA projection of extractive summaries chosen by multiple aspects of algorithms (`CNNDM`, `NewsRoom`, `XSum`, `PeerRead`, and `PubMed`).



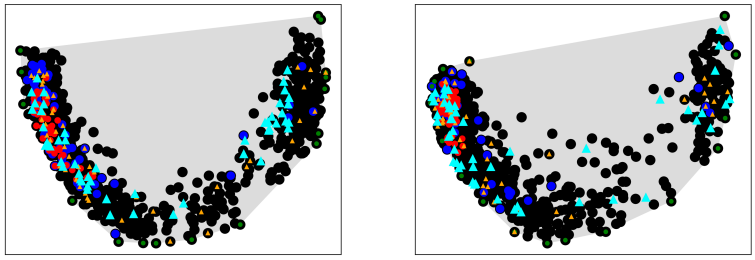
(a) PubMed



(b) Reddit



(c) AMI

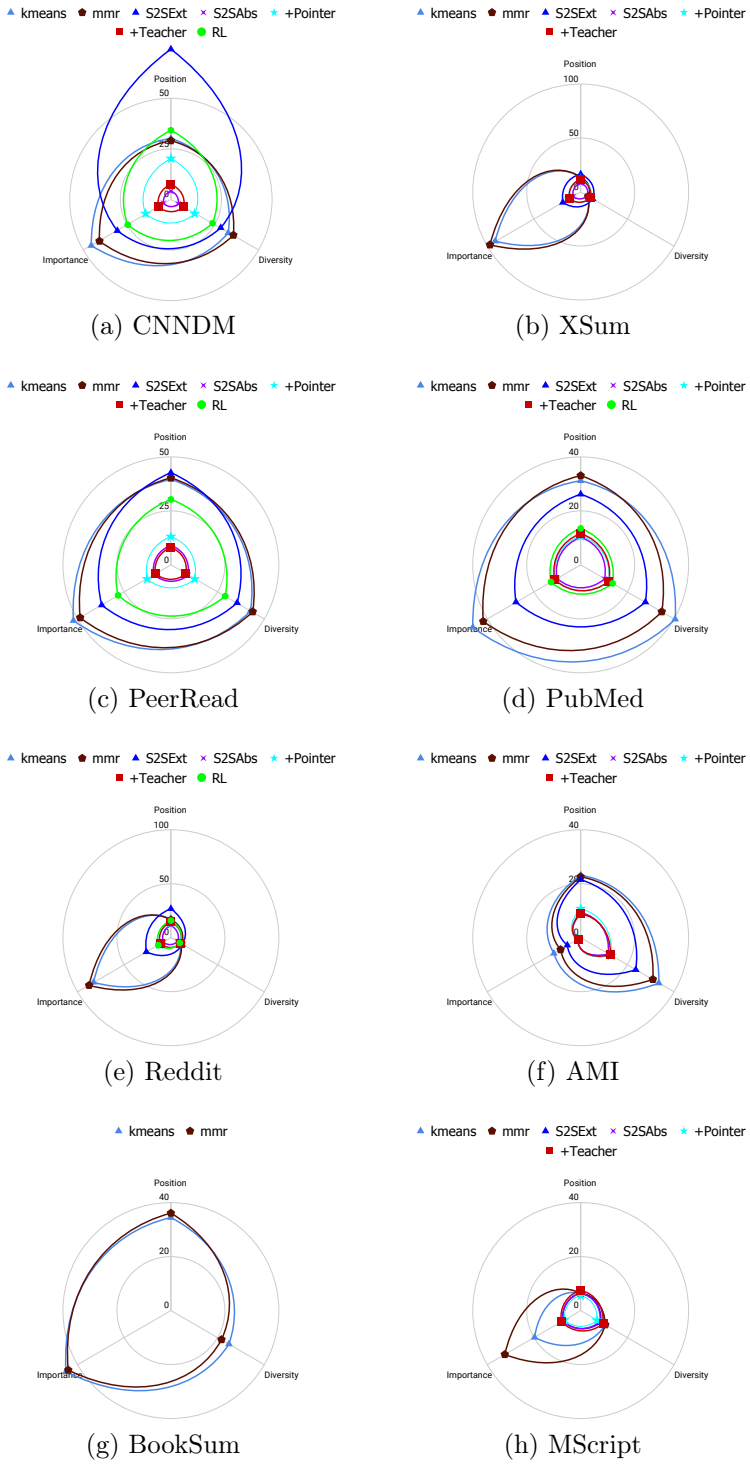


(d) BookSum

Figure 21: PCA projection of extractive summaries chosen by multiple aspects of algorithms (`Reddit`, `AMI`, `Booksum`, and `MScript`).

Appendix E System Biases per each corpus with the Three Sub-aspects

In Figure 22, we have more diagrams showing system biases toward each of three sub aspects. We find that there exists a bias according to the corpus: for example in `Reddit`, many systems have a importance bias in common. On the other hand, systems are biased toward a diversity aspect in `AMI`. Also, some systems tend to be biased in certain aspect across the different corpus: systems such as *KMeans* and *MMR*, many corpora are biased toward a importance aspect.



(a) CNNDM (b) XSum

(c) PeerRead (d) PubMed

(e) Reddit (f) AMI

(g) BookSum (h) MScript

Figure 22: System biases with the three sub-aspects per each corpus, showing what portion of aspect is used for each system.

Appendix F Details on Hyper-Parameters for PosCal

All models are trained with equal hyper-parameters: learning rate 2e-5, and BERT model size BERT_{BASE}. Also, we set up an early stopping rule for train: we track the validation loss for every 50 steps and then halt to train if current validation loss is bigger than the averaged 10 prior validation losses (i.e., patience 10). For **L1**, we use the regularization weight value 1-e8. For **PosCal** , we set up another weight value λ for $\mathcal{L}_{Cal}$, and the number of updating empirical probability per epoch (u). We tune these two hyper-parameters per each task. For more details, see Table 14. As a baseline of post-calibration method, we also report ECE with a temperature scaling [Guo et al., 2017], which is current state-of-the-art method.

xSLUE	u	λ	**GLUE**	u	λ
GYAFC	5	0.6	CoLA	5	0.2
SPolite	5	0.6	SST-2	10	1.0
SHumor	5	1.0	MRPC	10	1.0
SJoke	5	1.0	QQP	10	1.0
SarcGhosh	5	0.6	MNLI	2	0.2
SARC	5	0.6	MNLI_{mm}	2	0.2
SARC_pol	5	1.0	QNLI	1	0.6
VUA	2	1.0	RTE	10	1.0
TroFi	5	1.0	WNLI	2	0.2
CrowdFlower	5	0.6			
DailyDialog	5	1.0			
HateOffens	5	1.0			
SRomance	5	1.0			
SentiBank	5	1.0			
PASTEL_gender	5	1.0			
PASTEL_age	5	1.0			
PASTEL_country	5	1.0			
PASTEL_politics	5	1.0			
PASTEL_education	5	1.0			
PASTEL_ethnics	5	1.0			

Table 14: Hyper-parameters for **PosCal** training across tasks : the number of updating empirical probabilities per epoch u and weight value λ for $\mathcal{L}_{Cal}$. We tune them using the validation set.

Appendix G Examples When MLE and PosCal Predicts Different Label

Table 15 and 16 shows some examples in StanfordPoliteness and RTE datasets with their predicted $\hat{p}$ of true label from **MLE** and **PosCal** .

Data	Sentence	True label	MLE $\hat{p}$	PosCal $\hat{p}$
SPolite.	I don't know what page you are talking about, as this is your only edit. Did you perhaps have another account?	impolite	47.3	65.4 INCOR → COR
	Hi. Not complaining, but why did you remove the category "high schools in california" from this article?	impolite	1.2	91.7 INCOR → COR
	Hi, sorry I think I'm missing something here. Why are you adding a red link to the vandalism page?	impolite	5.6	61.9 INCOR → COR
	Can you put an NSLog to make sure it's being called only once? Also, can you show us where you are declaring your int?	polite	16.5	76.5 INCOR → COR
	I don't understand the reason for <url>. Would you please explain it to me?	polite	91.5	37.1 COR → INCOR
	Another question: Does "Senn" exist in Japanese? If it does, is it possible to render Sennin as Senn-in?	polite	88.8	45.5 COR → INCOR
	@Smjg, thanks. But why did you also remove the categories I added?	impolite	78.3	45.7 COR → INCOR
	You can place islands so there is no path between points. What should happen then?	impolite	91.7	35.8 COR → INCOR

Table 15: Predicted $\hat{p}(\%)$ of true label from **MLE** and **PosCal** with corresponding sentences in Stanford's politeness (bottom) dataset.

Data	Sentence	True label	MLE $\hat{p}$	PosCal $\hat{p}$
RTE	(S1) Charles de Gaulle died in 1970 at the age of eighty. He was thus fifty years old when, as an unknown officer recently promoted to the (temporary) rank of brigadier general, he made his famous broadcast from London rejecting the capitulation of France to the Nazis after the debacle of May-June 1940. (S2) Charles de Gaulle died in 1970.	entail	34.9	58.9 INCOR → COR
	(S1) Police in the Lower Austrian town of Amstetten have arrested a 73 year old man who is alleged to have kept his daughter, now aged 42, locked in the cellar of his house in Amstetten since 29th August 1984. The man, identified by police as Josef Fritzl, is alleged to have started sexually abusing his daughter, named as Elisabeth Fritzl, when she was eleven years old, and to have subsequently fathered seven children by her. One of the children, one of a set of twins born in 1996, died of neglect shortly after birth and the body was burned by the father. (S2) Amstetten is located in Austria.	entail	45.5	57.3 INCOR → COR
	(S1) Blair has sympathy for anyone who has lost their lives in Iraq. (S2) Blair is sorry for anyone who has lost their lives in Iraq.	entail	31.3	50.1 INCOR → COR
	(S1) The U.S. handed power on June 30 to Iraqâs interim government chosen by the United Nations and Paul Bremer, former governor of Iraq. (S2) The United Nations officially transferred power to Iraq.	not entail	59.2	44.9 COR → INCOR

Table 16: Predicted $\hat{p}(\%)$ of true label from **MLE** and **PosCal** with corresponding sentences in RTE dataset.

Appendix H Calibration confidence interval with bootstrap sampling

In this section, we describe how to apply a bootstrap confidence interval from Hall and Horowitz [2013] to the calibration probability estimates. Following is a simple note for our application.

- Fit the model $Y_i = g(X_i) + \epsilon_i$. In our case, X_i is predicted probability from original model and g is `KNN` with $k = n^{\frac{2}{3}}$. Here, calculate a variance of original set's residual $\hat{\sigma} = \frac{1}{n}\sum(Y_{[i+1]} - Y_{[i]})^2$ where $Y_{[i]}$ is corresponding response of order statistic $X_{[i]}$.
- Get residuals by $[\hat{\epsilon}_i : \tilde{\epsilon}_i - \bar{\epsilon}]$ where $\bar{\epsilon}$ is a mean of $\tilde{\epsilon}_i$. This will be used to get residual bootstrap sample in next.
- Get bootstrap sample by keeping same X_i but modify $Y_i^* = Y_i + \hat{\epsilon}$ where $\hat{\epsilon}$ is randomly selected in $[\hat{\epsilon}_i : \tilde{\epsilon}_i - \bar{\epsilon}]$ where $\bar{\epsilon}$ with replacement.
- For each bootstrap sample, calculate bootstrap version of $\hat{g}$, $\hat{\sigma}$ and bounds B where it covers 95% of C.I for all bootstrap samples: $[\hat{g}(x) - S(X)(x)\hat{\sigma}z_{1-\frac{\alpha}{2}}, \hat{g}(x) + S(X)(x)\hat{\sigma}z_{1-\frac{\alpha}{2}}]$. Since our $g(x)$ is a simple `KNN`, here $S(X)(x)$ is just $\sqrt{\frac{1}{k}}$, following the fact that $S(X)(x)\sigma$ refers to the variance of `KNN` estimators.
- Get empirical coverage rates from all bootstrap samples. Choose our optimal α as α_0 quantile over all of this coverage rates. According to the paper, $\alpha_0 = 0.9$ generally works well so we also apply this.
- Get final C.I with original set and original σ based on the optimal α_0 above.

Bibliography

Miltiadis Allamanis, Hao Peng, and Charles Sutton. A convolutional attention network for extreme summarization of source code. *arXiv preprint arXiv:1602.03001*, 2016.

Imanol Arrieta-Ibarra, Paman Gujral, Jonathan Tannen, Mark Tygert, and Cherie Xu. Metrics of calibration for probabilistic predictions. *arXiv preprint arXiv:2205.09680*, 2022.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. Multi-document abstractive summarization using ilp based multi-sentence compression. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2015.

C Bradford Barber, David P Dobkin, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4): 469–483, 1996.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 481–490. Association for Computational Linguistics, 2011.

Gérard Biau and Luc Devroye. *Lectures on the nearest neighbor method*, volume 246. Springer, 2015.

Peter J Bickel and Anat Sakov. On the choice of m in the m out of n bootstrap and confidence bounds for extrema. *Statistica Sinica*, pages 967–985, 2008.

Steinar Bjerve, Kjell A Doksum, and Brian S Yandell. Uniform confidence bounds for regression based on a simple moving average. *Scandinavian journal of statistics*, pages 159–169, 1985.

James Booth, Peter Hall, and Andrew Wood. Bootstrap estimation of conditional distributions. *The Annals of Statistics*, pages 1594–1610, 1992.

Florian Boudin and Emmanuel Morin. Keyphrase extraction for n-best reranking in multi-sentence compression. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013.

Florian Boudin, Hugo Mougard, and Benoit Favre. Concept-based summarization using integer linear programming: From concept pruning to multiple optimal solutions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2015*, 2015.

Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

Dallas Card and Noah A Smith. The importance of calibration for estimating proportions from annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1636–1646, 2018.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. The ami meeting corpus: A pre-announcement. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 28–39. Springer, 2005.

George H Chen, Devavrat Shah, et al. *Explaining the success of nearest neighbor methods in prediction*. Now Publishers, 2018.

Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*, 2016.

A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5 2:153–163, 2017.

Trevor Cohn and Mirella Lapata. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics, 2008.

Ernest Cooke. Forecasts and verifications in western australia. *Monthly Weather Review*, 34 (1):23–24, 1906.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Timo Dimitriadis, Lutz Duembgen, Alexander Henzi, Marius Puke, and Johanna Ziegel. Honest calibration assessment for binary outcome predictions. *arXiv preprint arXiv:2203.04065*, 2022.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. Learning-based single-document summarization with compression and anaphoricity constraints. *arXiv preprint arXiv:1603.08887*, 2016.

Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

Randall L Eubank and Paul L Speckman. Confidence bands in nonparametric regression. *Journal of the American Statistical Association*, 88(424):1287–1301, 1993.

Katja Filippova. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. Association for Computational Linguistics, 2010.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*, 2018.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. Abstractive summarization of product reviews using discourse structure. In *Proceedings of EMNLP*, 2014.

Dan Gillick and Benoit Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18. Association for Computational Linguistics, 2009.

Achintya Gopal. Why calibration error is wrong given model uncertainty: Using posterior predictive checks with deep learning. *arXiv preprint arXiv:2112.01477*, 2021.

Philip John Gorinski and Mirella Lapata. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, 2015.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34, 2003.

Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL http://aclweb.org/anthology/N18-1065.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Peter Hall. On bootstrap confidence intervals in nonparametric regression. *The Annals of Statistics*, pages 695–711, 1992.

Peter Hall and Joel Horowitz. A simple bootstrap method for constructing nonparametric confidence bands for functions. *The Annals of Statistics*, pages 1892–1921, 2013.

Wolfgang Härdle and Enno Mammen. Bootstrap methods in nonparametric regression. In *Nonparametric functional estimation and related topics*, pages 111–123. Springer, 1991.

Trevor Hastie, Robert Tibshirani, and Ryan J Tibshirani. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*, 2017.

Kai Hong and Ani Nenkova. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, 2014.

David W Hosmer and Stanley Lemesbow. Goodness of fit tests for the multiple logistic regression model. *Communications in statistics-Theory and Methods*, 9(10):1043–1069, 1980.

Taehee Jung, Dongyeop Kang, Lucas Mentch, and Eduard Hovy. Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3324–3335, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1327. URL https://aclanthology.org/D19-1327.

Taehee Jung, Dongyeop Kang, Hua Cheng, Lucas Mentch, and Thomas Schaaf. Posterior calibrated training on sentence classification tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2723–2730, 2020.

Dongyeop Kang and Eduard H. Hovy. xslue: A benchmark and analysis platform for cross-style language understanding and evaluation. *ArXiv*, abs/1911.03663, 2019.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. A dataset of peer reviews (peerread): Col-

lection, insights and nlp applications. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, New Orleans, USA, June 2018. URL https://arxiv.org/abs/1804.09635.

Ramakanth Kavuluru, Anthony Rios, and Y. Lu. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine*, 65 2:155–66, 2015a.

Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine*, 65(2):155–166, 2015b.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. Content selection in deep learning models of summarization. *arXiv preprint arXiv:1810.12343*, 2018.

Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12295–12305, 2019.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, pages 3787–3798, 2019.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8, 2004.

Chin-Yew Lin and Eduard Hovy. Identifying topics by position. In *Fifth Conference on Applied Natural Language Processing*, 1997.

Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics, 2000.

Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920. Association for Computational Linguistics, 2010.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

Hui Lin and Jeff A Bilmes. Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*, 2012.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, 2015.

Xingchen Ma and Matthew B Blaschko. Meta-cal: Well-controlled post-hoc calibration by ranking. In *International Conference on Machine Learning*, pages 7235–7245. PMLR, 2021.

Daniel Marcu. Discourse trees are good indicators of importance in text. *Advances in automatic text summarization*, 293:123–136, 1999.

Ryan McDonald. *A study of global inference algorithms in multi-document summarization*. Springer, 2007.

Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proc. of ACL*, pages 1220–1230, 2014.

Rada Mihalcea and Hakan Ceylan. Explorations in automatic book summarization. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

Robert G Miller. Statistical prediction by discriminant analysis. In *Statistical Prediction by Discriminant Analysis*, pages 1–54. Springer, 1962.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600, 1973.

Allan H Murphy and Robert L Winkler. Probability forecasting in meteorology. *Journal of the American Statistical Association*, 79(387):489–500, 1984.

M. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, 2015.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, 2016.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*, 2018a.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*, 2018b.

Giovanni Nattino, Stefano Finazzi, and Guido Bertolini. A new calibration test and a reappraisal of the calibration belt for the assessment of prediction models based on dichotomous outcomes. *Statistics in medicine*, 33(14):2390–2407, 2014.

Michael H Neumann and Jörg Polzehl. Simultaneous bootstrap confidence bands in nonparametric regression. *Journal of Nonparametric Statistics*, 9(4):307–333, 1998.

Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1587–1598, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1182. URL https://www.aclweb.org/anthology/D15-1182.

Khanh Nguyen and Brendan T. O'Connor. Posterior calibration and exploratory analysis for natural language processing models. In *EMNLP*, 2015.

Chenri Ni, Nontawat Charoenphakdee, Junya Honda, and Masashi Sugiyama. On the calibration of multiclass classification with rejection. In *Advances in Neural Information Processing Systems*, pages 2582–2592, 2019.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, volume 2, 2019.

Jessica Ouyang, Serina Chang, and Kathy McKeown. Crowd-sourced iterative annotation for narrative summarization corpora. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 46–51, 2017.

Prabasaj Paul, Michael L Pennell, and Stanley Lemeshow. Standardizing the power of the hosmer–lemeshow goodness of fit test in large data sets. *Statistics in medicine*, 32(1): 67–80, 2013.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Maxime Peyrard. A simple theoretical model of importance for summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1059–1073, Florence, Italy, July 2019a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1101.

Maxime Peyrard. Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy, July 2019b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1502.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Dimitris N Politis. Scalable subsampling: computation, aggregation and inference. *arXiv preprint arXiv:2112.06434*, 2021.

Dimitris N Politis and Joseph P Romano. Large sample confidence regions based on subsamples under minimal assumptions. *The Annals of Statistics*, pages 2031–2050, 1994.

Dimitris N Politis, Joseph P Romano, and Michael Wolf. *Subsampling*. Springer Science & Business Media, 1999.

Dimitris N Politis, Joseph P Romano, and Michael Wolf. On the asymptotic theory of subsampling. *Statistica Sinica*, pages 1105–1124, 2001.

Rebecca Roelofs, Nicholas Cain, Jonathon Shlens, and Michael C Mozer. Mitigating bias in calibration error estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 4036–4054. PMLR, 2022.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association

for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://www.aclweb.org/anthology/W18-5446.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. *Advances in Neural Information Processing Systems*, 32, 2019.

Kristian Woodsend and Mirella Lapata. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the conference on empirical methods in natural language processing*, pages 409–420. Association for Computational Linguistics, 2011.

Fan Yang and Rina Foygel Barber. Contraction and uniform convergence of isotonic regression. *Electronic Journal of Statistics*, 13(1):646–677, 2019.

Dani Yogatama, Fei Liu, and Noah A Smith. Extractive summarization by maximizing semantic volume. In *EMNLP*, pages 1961–1966, 2015.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 609–616, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655658.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM, 2002.

David Zajic, Bonnie Dorr, and Richard Schwartz. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, pages 112–119, 2004.

Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning*, pages 11117–11128. PMLR, 2020.