Learning with Data Sampling Biases for Natural Language Understanding

Anonymous ACL submission

Abstract

In recent years, NLP models have dramatically improved by utilizing user data, enabling commercial products such as chat bots and smart voice agents. However, data collected for training such models may suffer from sampling biases, conditioned on the dataset collection protocol. Additionally, a practitioner may not always obtain datasets of the desired volumes, particularly given the emerging privacy considerations (e.g. relying on a user to donate their data for model-building purposes). In this paper, we simulate various scenarios under which one may obtain biased training datasets for the task at hand. We build baselines simulating various biased data sampling conditions and present observations such a biased data collection that obtains data-points away from class centroids offer more value. We also test two sets of data augmentation algorithms: (i) pseudo-labeled data through semi-supervised learning, assuming availability of unlabeled data and, (ii) data augmentation through synthetic data generation. We observe that while the best performing data augmentation method depends on the biased setting and the dataset, simple data augmentation algorithms (such as *Easy Data Augmentation*) are still largely effective.

1 Introduction

Data collection is an integral part of training any ML system and the data collection protocol can significantly impact the performance of the ML model. While, arguably, an unrestricted access to the data source for unbiased data collection in large volumes is desirable, it may not always be the case. For example, under certain conditions, data collection protocols may dictate separate data collection per label of interest (e.g., requesting a study group to generate variants of music request to build a spoken language understanding model, which otherwise also supports other non-music requests). Similarly, data collection may be restricted to offer only a biased sub-sample of the data (e.g., in another scenario, while building a spoken language understanding system, a biased section of user population may donate their data). Additionally, gathering labeled data in large volumes may not always be feasible given increasing emphasis on user data privacy. In this work, we study the impact of such biases introduced during the dataset collection protocol on the model performance.

Researchers have investigated biases in training datasets (Tommasi et al., 2015), and its impact on the model performance. However, impact of various types of sampling biases in NLU modeling is not well studied. Particularly, given current advances in NLU modeling, where task-specific models are fine-tuned on top of pre-trained models, the impact of sampling biases has not been evaluated.

We simulate settings that mimic different kinds of biases that can be introduced during data collection. In addition to a random downsampling, our simulations introduce biases under data collection protocols that either collect data independently the supported set of labels or, collect data for all the labels together. Furthermore, we simulate these biases in a low data volume setup when only tens or hundreds of data-points are available for each class. We focus on biases in low data settings as the impact of biases is expected to be more pronounced and, low availability of data is an increasing realistic scenario in building industrial ML systems given emerging privacy considerations (Bender and Friedman, 2018). Furthermore, we benchmark two sets of data augmentation methods: (i) semi-supervised learning assuming availability of unlabeled data and, (ii) synthetic data generation, to assess their value in recovering from low-data and biased training data. We discuss observations such as while the best performing data augmentation method is a function of the bias setting, simple method such as *Easy Data Augmentation* (Wei and Zou, 2019) generally perform well.

2 Related Works

2.1 Bias in Dataset Collection

The quality and real-world utility of datasets used to train and evaluate machine learning models is highly sensitive to biases in the processes used to create them (Bender and Friedman, 2018). Bias can appear in all parts of the dataset-creation pipeline, including the curation methods used to select which examples to include in a dataset (Zhou et al., 2021; Tommasi et al., 2015), the design of the annotation guidelines and prompts (Schwartz et al., 2017), the subjective judgements made by individual annotators (Geva et al., 2019; Wich et al., 2020; Gururangan et al., 2018), and the decisions about how to split a dataset into training, validation, and test sets (Zhou et al., 2021). Models trained on these biased datasets may then learn to exploit dataset-specific artifacts (Gururangan et al., 2018; Tsuchiya, 2018), achieving strong performance on similarly-biased test sets, but not generalizing well to other examples from the task's real-world data distribution.

In recent years, there have been many related efforts to mitigate the effects of these hidden dataset biases through improved dataset creation and annotation procedures (Geva et al., 2019; Schwartz et al., 2017; Wich et al., 2020; Zhou et al., 2021; Stasaski et al., 2020; Bender and Friedman, 2018), data augmentation methods (Zhou and Bansal, 2020; Park et al., 2018; Min et al., 2020; Shinoda et al., 2021), and bias-aware learning algorithms (Jiang and Nachum, 2020; Clark et al., 2020; He et al., 2019; Li and Vasconcelos, 2019; Khosla et al., 2012; Zhao et al., 2017). In this work, we propose novel methods to create biased datasets from existing, publicly-available datasets through selective downsampling. We then use these methods to 1) create several benchmark text classification datasets with different types of bias; 2) evaluate the performance of several techniques to mitigate these biases, including semi-supervised learning (Ouali et al., 2020), off-the-shelf data augmentation techniques (Wei and Zou, 2019), and paraphrase generation with large language models (Witteveen and Andrews, 2019). We further elaborate on the state of research in data augmentation methods used in this paper below.

Semi-supervised learning In many ML applications, it is relatively easy to collect unlabeled data points from public sources such as the Internet, while high quality human labels are harder and more expensive to obtain in large scale (Zhu, 2005). In these cases, semi-supervised learning (Van Engelen and Hoos, 2020) is a commonly employed strategy where a large unlabeled set of data samples are used along with a small labeled set. The unlabeled data can be used either in pre-training, as a part of the training objective, or by generating new *pseudo-labels* for the unlabeled samples, followed by direct augmentation to the training data (Van Engelen and Hoos, 2020). Of these, pseudo-labeling (Lee et al., 2013) is considerably simple as it needs minimal changes to existing training routines, and is frequently used in literature (Triguero et al., 2015). Generating the labels can be done using a *seed model* initially trained only on the labeled dataset, or by clustering the labeled and unlabeled samples and assigning majority labels obtained from the labeled examples. In this work, we experiment with both strategies.

Data generation by distorting existing data This form of augmentation is commonly applied in computer vision where images or frames are cropped, flipped or their RGB channels suitably noised. However, simple alterations such as these may not translate well to NLP and have been reported to create meaningless utterances (Liu et al., 2020). More recent works instead try to generate new data by introducing word level changes (Kobayashi, 2018; Wang and Yang, 2015), by generating semantically similar paraphrases (Gupta et al., 2018a), or by employing large language models such as GPT-2 to generate new utterances (Liu et al., 2020). Easy Data Augmentation (EDA) (Wei and Zou, 2019) introduces word level distortions and includes four simple operations (synonym replacement, random insertion, swap and deletion) to generate new data, and has found considerable acceptance due to its simplicity. In this work, we experiment with both EDA and paraphrase based data augmentations to generate new data.

3 Creating datasets with sampling biases

Conditioned on the dataset collection protocol or other aforementioned factors, different biases may creep into the obtained data. We discuss three such scenarios below.

Scenario 1: Unbiased data collection. In this scenario, the practitioner is capable of sampling data from the real world distribution. This scenario is likely, for example, when the practitioner has unrestricted access to the process governing data generation.

Scenario 2: Biased data collection per-class. In certain scenarios, practitioners are obligated to gather data per class. For example, in an industrial setting, one may launch ML models with a pre-defined class support (e.g. a model that classifier utterances into PlayMusicIntent and GetWeatherIntent). To launch models with the given class support, the practitioner may be required to collect representative utterances per class (by requesting paid users to make either requests to play music or get weather to get coverage for PlayMusicIntent and GetWeatherIntent, respectively). The distribution of such utterances within each class, however, may not conform to the real-world distribution.

Scenario 3: Biased data collection across classes. In this scenario, the practitioner first collects data for the pre-defined class support and then trains a model on the collected data. However, they are not able to collect data as per the real world distribution. For example, given the full class support, the practitioners may only be able to get representative datapoints from a set of users who agree to donate their data.

We further introduce operating with reduced data volumes in all the scenarios above as motivated earlier. We also note that we enforce that at least one data point is available per class in each simulation. This is important as unconstrained severe undersampling may lead to a reduced class support, as datapoints from some classes may not be sampled. We discuss our setup for simulating above scenarios in the next section.

3.1 Simulating dataset collection

Motivated by the aforementioned scenarios, we discuss simulations to mimic them below.

Scenario 1: Uniform random down-sampling. In this method, we randomly downsample the available dataset to a fraction of its original size. This method is expected to provide a smaller number of datapoints available, but does not introduce any bias in the sampled data.

Scenario 2: Class dependent bias injection. In this bias injection method, we under-sample datapoints per class. In particular, when requesting a set of users to generate datapoints specific to a class, they may tend to produce similar set of requests (e.g. given a task to generate data for PlayMusicIntent, a user may provide pop music requests, while another user may provide classical music requests). Using this as a motivation, given a class, we obtain K seed datapoints from amongst the datapoints belonging to that class. Given the seed datapoints, we select utterances proximal to the seeds (as defined through a chosen embedding space) to obtain the undersampled data. Following the example above, each seed can be seen as a prototype of requests a user makes and the proximal utterances can be expected to provided by the same user.

We propose multiple ways of selecting the seed datapoints. In our experiments, we use the following settings: (i) $K = 1$, seed close to class centroid, (ii) $K = 1$, seed away from class centroid, (iii) $K > 1$ seeds away from class centroid and, (iv) $K > 1$, seeds randomly chosen. The class centroid is again computed based on all the available datapoins for the class at hand, as defined on the chosen embedding space.

Scenario 3: Class agnostic bias injection. In this method, we obtain K seed datapoints and select utterances proximal to the seed datapoint without factoring in the class assignments. This leads to semantically similar utterances finding prevalence in the under-sampled data, without considering the class. This dataset creation mechanism mimics a scenario where a biased set of datapoints are selected from the real distribution, which are then annotated for class labels for training a classifier.

For each of the methods described above, we operate in an utterance embedding space computed based on the smooth inverse frequency (SIF) method (Sanjeev Arora, 2017). SIF embeddings have been shown as a strong, yet simple method to obtain sentence embeddings. We select seed utterances in the SIF embeddings space and select proximal utterances based on the L2 norm. We also note that in the real world the process for biased data generation is unlikely to be available to the modeler. Therefore, we do not use SIF based embeddings in any of our methods to benchmark improvements on the biased data samples. We show crafted visual demonstrations of the simulations for selected scenarios in the Figure 1.

3.2 Datasets used

We use three English datasets for our experiments, as summarized below.

ATIS Intent Classification Dataset (Chen,

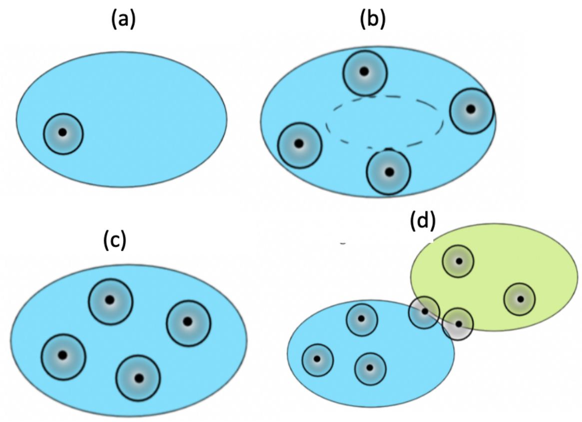

Figure 1: This figure demonstrates sampling the data under different bias settings. Assuming the span of a chosen class is shown using the blue ellipse, (a) shows sampling with a single seed ($K = 1$) with the seed selected away from the class centroid. Similarly, (b) shows sampling with multiple seeds ($K > 1$) with seeds away from centroid. (c) shows sampling with several randomly selected seeds, and (d) shows sampling with seeds selected randomly irrespective of the class (green ellipse denotes a class separate to the one denoted by the blue ellipse).

2019): This dataset consists of 4952 utterances in training set and 878 in test set, split across 18 intents.

Semantic Parsing for Task Oriented Dialog using Hierarchical Representations (TOP) (Gupta et al., 2018b): TOP contains 31279 utterances in the training set and 9042 in test set, across 19 intents.

SNIPS Natural Language Understanding benchmark (Alice Coucke, 2018): SNIPS contains 13784 utterances in the training set and 700 in test set, across 7 intents.

3.3 Performance baselines

Given the created datasets, we train intent classifiers on them and report our findings in Table 1. For the random down-sampling, we obtain datasets sized to 1%, of its original volume (we report numbers on sampling 5% and 10% of the data in the Appendix X). We continue selecting nearest utterances to the selected seed utterances until we cover 1% of the overall data volume (same heuristic is applied for sampling 5% and 10% of the traffic). We fine-tune a BERT base model(110M parameters) on the available labeled data for all our classification tasks. We create 10 versions of datasets in biased setting and present average performance across them.

Setting	ATIS	TOP	SNIPS
Random down-sampling, 1% data			
Random	66.52%	83.50%	85.81%
Class dependent bias injection, 1% data			
($K = 1$ close to centroid)	70.59%	73.45%	68.51%
($K = 1$ away from centroid)	72.30%	72.22%	75.22%
($K > 1$ away from centroid)	80.77%	77.65%	80.77%
($K > 1$)	73.69%	74.39%	75.04%
Class independent bias injection, 1% data			
($K > 1$)	72.21%	72.76%	34.40%

Table 1: Baseline results, trained with 1% labelled data

3.4 Observations

We discuss various observations on the baseline performances below.

1. **While random down-sampling performs the best in TOP and SNIPS, it is the worst performing baseline in ATIS.** We expected that random down-sampling to perform the best given that it preserves class distribution across data-samples. However, this is not the case in the ATIS dataset sampled down to 1% of its size. We identify that in a few shot learning scenario, it is hard to sample data that matches the true distribution. Severe under-sampling in ATIS leaves room for 1-2 samples per class, as shown in Table 2. We also observe that gathering biased data per-class yields more samples for under-represented classes (e.g. capacity/distanc), leading to better accuracy. This implies that during few shot learning, it is better to have more representative data-points from each class, as opposed to a more matched class distribution. We observe that as the number of random samples increase (from 1% to 10%), the performance of random baseline improves (please see Appendix for numbers on datasets with size 5% and 10%).

2. **($K > 1$ away from centroid) performs the best in biased settings.** We observe that gathering diverse set of data per-class that is distant from class centroid yield the most value in terms of determining class boundaries. Datapoints away from centroid are more likely to be close to the decision boundary and data sampling methods such as active learning rely on a similar heuristic

Intent/Ratio	10%	1%	10%	1%
abbreviation	11	2	12	3
aircraft	8	1	9	2
airfare	41	5	42	6
airline	15	2	16	3
airport	2	1	3	2
capacity	2	1	3	2
cheapest	1	1	1	1
city	2	1	4	2
distance	3	1	4	2
flight	343	35	340	30
flight_no	2	1	4	2
flight_time	6	1	7	2
ground_fare	2	1	3	2
ground_service	24	3	25	4
meal	1	1	2	2
quantity	5	1	6	2
restriction	1	1	2	2

Table 2: Number of Utts in each intent of Atis with random sampling

to gather valuable annotated data.

3. **The class independent bias injection setting** $(K > 1)$ **severely under-performs for SNIPS.** We observe an average performance of 34.4% for class independent bias injection in SNIPS (we emphasize that this performance is average across 10 samples of the data and thus, not a one off observation). However, we observe a good recovery in case of using 5% or 10% of the data (results in Appedix X). We show the number of datapoints per class 1% and 10% data volume setting for random down sampling and a biased sampling setting in Table 3 (sampled from one of the 10 versions). We observe that severe under-sampling in SNIPS leads to a skew in the training data with intents like 'GetWeather'/'SearchScreeningEvent' observing far fewer datapoints (note that these classes otherwise are fairly frequent as seen in 10% and 5% sampled data). [Check this] This is due to the fact that this intent while very frequent are tightly clustered in the embedding space. If a seed is not chosen close to the cluster, they are likely to be severely under-represented. In a real world setting, this setting is analogous to a case where a very similar set of users may provide most data for a frequent class, but they refrain from donating their data.

Table 4 shows the skewed distribution caused by cross intent biased sampling in Snips, which originally has same amount data within each intent.

Intent/Ratio	10%	5%	1%
AddToPlaylist	28	11	10
BookRestaurant	396	137	79
GetWeather	234	164	2
PlayMusic	50	20	1
RateBook	283	191	36
SearchCreativeWork	83	13	11
SearchScreeningEvent	290	147	1

Table 3: Number of Utts in each intent of Snips with cross intent biased sampling

4 Methods for Benchmarking

Given the methods to generate under-sampled datasets as described above, we benchmark two broad categories of data augmentation methods on each baseline: (i) Data augmentation through semi-supervised learning and, (ii) Data augmentation through data generation. We describe them below. (all the computing works take around 1 week of an AWS p3 instance, with 8 nVidia Tesla V100)

4.1 Semi Supervised Learning

In this setting, we assume availability of unlabeled datapoints for the dataset at hand. Furthermore, we assume that the available unlabeled data follows the real world distribution. We then use two ways of label-propagation on the unlabeled data to generate pseudo-labeled data. The pseudo-labeled data is then augmented with the labeled data to train a classifier. We expect that the unlabeled data that follows the real distribution can correct for biases in the labeled data.

4.1.1 Self-learning based SSL

In this method, we train a seed model on the labeled data and pseudo-label the unlabeled data with the seed model. For both, the seed and the model trained on augmented data, we use a BERT based pre-trained model trained from ConSert and fine-tune it on the labeled data.

4.1.2 Clustering based SSL

In this method, we propagate labels from the labeled datapoints to neighboring un-labeled datapoints. Similar to (Aharoni and Goldberg, 2020), we use a pre-trained LM to first produce sentence embeddings for both labeled and unlabeled datapoints. The unlabeled data helps the model to learn the overall data pattern in the dataset while the labeled data helps the model to label the unlabeled data. Our proposed method runs clustering with large amount of unlabelled data and only select the most confident clusters to ensure the quality of pseudo-labels. We summarize the steps used in this method below: (i) We first use an LM (BERT) to obtain sentence representations. (ii) We use K-means clustering on the LM representations obtained for labeled and unlabeled data to identify clusters. We expect that each cluster represents a set of semantically similar sentences. To ensure fine granularity of clustering, the number of clusters is set much larger than the number of classes (e.g., number of domains or intents) (Mahon and Lukasiewicz, 2021). (iii) We then pseudo-label unlabeled datapoints in selected clusters based on the set of labeled datapoints in the cluster. Recent work showed that pseudo-labels perform poorly mainly because of low accuracy in clustering (Divam Gupta and Sivathanu, 2020). Consequently, similar to (Ishii, 2021), we only keep the most "pure" clusters, as we define next. A pure cluster has the following properties (a) At least 1% of the datapoints in a given cluster need to be labeled, (b) the majority class amongst the labeled datapoints needs to account for at least 80% of the labeled datapoints. All unlabeled datapoints in each pure cluster is assigned the label same as the majority class in the respective cluster. Once a set of unlabeled datapoints are pseudo-labeled, we train a classifier on the combined set of labeled and pseudo-labeled data.

4.2 Data augmentation

In this setting, we assume that no unlabeled data is available for the task of interest and we focus on generating more data from the labeled data using the following set of methods.

4.2.1 Easy Data Augmentation

EDA (Wei and Zou, 2019) is a data augmentation technique that uses synonym replacement/ random synonym insertion/ random two words swap and random word removal to synthesize new training examples. It creates 9 generated utterances per labelled utterance using these four techniques. While the heuristic behind EDA is simple, it has shown to outperform several strong data generation baselines.

4.2.2 Back-translation

Back-translation (BT) (Sennrich et al., 2016) is a commonly used approach for paraphrasing text: a machine translation (MT) system is applied to translate text from the source language to a target pivot language, then back again. By using n-best in both directions, BT can produce a large number of paraphrases. We fine-tune an internal 5B parameter seq2seq model on WMT 2014 data(Bojar et al., 2014), using a single model for en→fr and fr→en, with an instruction prompt to control the language direction: "Translate to French:" and "Translate to English:", respectively. We decode with beam search using M=10 forward and N=10 backward translations, to produce up to 100 variations of each original sentence. After heuristic cleaning (removing invalid punctuation like "!" and "?.") and de-deduplication, the average number of outputs per input is 41 for ATIS, 51 for SNIPS, and 36 for TOP.

4.2.3 In-context Learning

Given the recent emergence of in-context learning as a way to generate quality data from large models, we use this as another baseline. We use a 20B parameter language model to generate data by setting the handful of labeled data for the task at hand as context. In particular, for each dataset and each intent, we give the model 3 utterances of that with a prompt (e.g., in the form `Example with [flight] intent: do you have an early morning direct flight from philadelphia to pittsburgh?`) and generate 27 samples of the same intent by letting the model continue generation after the final prompt(e.g.,`Example with [flight] intent:`). For generation, we use nucleus sampling with $p = 0.5, 0.7, 0.9$.

We augment various baselines discussed in Section 3 (that cover up to 1% of the training data) with data obtained through the semi-supervised learning and data augmentation methods (results for 5% and 10% settings are presented in Appendix). For SSL methods, we use data not selected during biased sampling as the unlabeled data. Same BERT-base architecture is used for fine-tuning on augmented datasets and the test set is consistent with the baselines presented in Section 3.3. Table 4 summarizes the results.

Dataset: ATIS								
Full data baseline	97.94							
	Baseline	SSL	Clustering	EDA	Gen_20Bp5	Gen_20Bp7	Gen_20Bp9	Gen_5B
Random down-sampling	66.5	68.1	78.4	82.4	83.6	85.8	87.3	82.5
Class dependent bias injection:								
($K = 1$ close to centroid)	70.6	70.4	50.3	80.2	77.7	76.9	78.5	78.9
($K = 1$ away from centroid)	72.3	72.8	46.8	78.7	79.1	80.9	83.7	75
($K > 1$ away from centroid)	76.5	81.5	58.8	84	84.7	86.3	85	83.2
($K > 1$)	76.7	77.6	52.5	80.5	82.4	85.4	86.8	81
Class independent bias injection:								
($K > 1$)	72.2	73	72.5	78.6	81	85.9	86.6	79.9
Dataset: Top								
Full data baseline	94.16							
	Baseline	SSL	Clustering	EDA	Gen_20Bp5	Gen_20Bp7	Gen_20Bp9	Gen_5B
Random down-sampling	83.5	83.8	83.8	86.9	84.5	84.6	84.4	87.5
Class dependent bias injection:								
($K = 1$ close to centroid)	73.5	74	59.3	75.7	67.2	69.9	73.8	75.4
($K = 1$ away from centroid)	72.2	72.6	56.8	74.5	70.9	72.9	74.6	73.8
($K > 1$ away from centroid)	77.3	78.1	69.4	80.6	73.2	75.6	78.5	78.9
($K > 1$)	74.9	77.8	63.3	77.8	73	76	79.4	80.1
Class independent bias injection:								
($K > 1$)	72.8	73.4	72.1	76	77.7	76.9	77.6	78.1
Dataset: Snips								
Full data baseline	98.86							
	Baseline	SSL	Clustering	EDA	Gen_20Bp5	Gen_20Bp7	Gen_20Bp9	Gen_5B
Random down-sampling	85.8	88.5	94	91.8	94.1	94.9	94.2	93.8
Class dependent bias injection:								
($K = 1$ close to centroid)	68.5	71.2	86.1	79.8	82.1	85.9	89.7	87.2
($K = 1$ away from centroid)	75.2	76.9	83	80.5	81.7	86.9	90.6	85.1
($K > 1$ away from centroid)	75.2	82.5	88	87.2	87.1	90.9	92	91
($K > 1$)	79.3	82.4	88.2	84.4	90	89.7	93.3	91.8
Class independent bias injection:								
($K > 1$)	34.4	33.9	73.5	47	56.1	69	69.5	57.4

Table 4: Performance(accuracy in test sets) of models, trained with 1% of labelled data and augmented data from each method

4.3 Observations

Examples of data generated through the data augmentation methods are shown in Table 5. We make the following observations from the results.

1. **Data generations methods are competitive to SSL methods** We observe that the data generation methods trained on top of models with large volumes of world knowledge (e.g. data from web crawl) or simple perturbations outperform models trained on a combination of labeled and pseudo-labeled data. We attribute this observation to the fact that semi-supervised techniques use for pseudo-labeling techniques are dependent on the seed set of labeled datapoints. In absence of a diverse and representative labeled datapoints, pseudo-labeling unlabelled data can be challenging.

2. **EDA emerges as a strong benchmark** Akin to the claims made in the EDA paper, we observe that their proposed method performs well in our baselines. The in-context based methods beats EDA in the class independent bias injection method, but otherwise EDA either beats or is fairly competitive.

3. **The clustering method yields value on the SNIPS dataset, while hurting the performance in other datasets.** While EDA and in-context learning generally perform the best, clustering based SSL outperforms other methods in SNIPS. We, therefore, analyze if a heuristic can capture when to select clustering based method. We look at T-SNE and identfy that there must be clean clusters. We also look at intra-cluster metric.

To analyze the reason behind the performance difference of the two pseudo labelling methods(SSL and clustering), we plot the t-SNE(van der Maaten and Hinton, 2008) embeddings of some

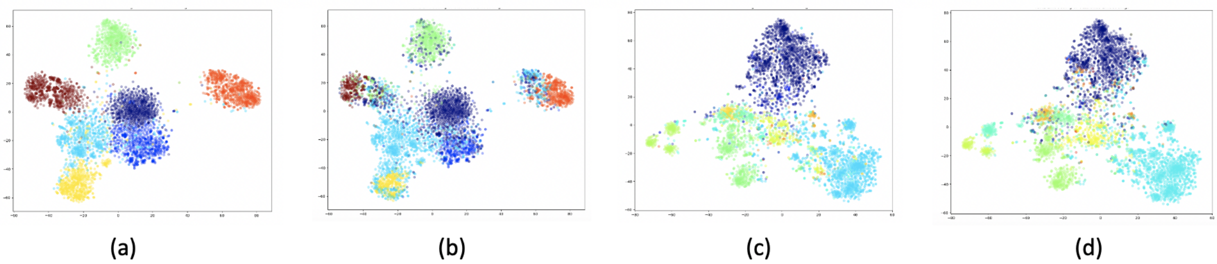

Figure 2: (a) Snips t-SNE with ground truth label (b) Snips t-SNE with ssl label (c) TOP t-SNE with ground truth label (d) TOP t-SNE with ssl label

random sampled utts from these dataset.

Figure 4 and 5 shows the situations in Snips, where clustering beats SSL. The color of embeddings in Figure 4 represents the ground truth label while in Figure 5 they are the pseudo label given by SSL. We can see even with well-clustered utts, SSL mis-labels a lot of them, SSL pseudo label accuracy is 68.9% for singled seeded sampling, 1% data retain rate, while in this setting, clustering has pseudo label accuracy of 87.1%.

However, as Figure 6 and 7 shows the situations in TOP, where clustering has lower accuracy compared with SSL. We can see in a dataset where the utts are not clustered well by intent, clustering cannot give a good help.

5 Conclusion

This survey gives an overview over data augmentation approaches to mitigate reduced annotation volumes and biased sampling for intent classification in different domains and dataset.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Adrien Ball Théodore Bluche Alexandre Caulier David Leroy Clément Doumouro Thibault Gisselbrecht Francesco Caltagirone Thibaut Lavril Maël Primet Joseph Dureau Alice Coucke, Alaa Saade. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *ArXiv*, abs/1805.10190.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Yun-Nung (Vivian) Chen. 2019. Atis intent classification dataset.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

Nipun Kwatra Divam Gupta, Ramachandran Ramjee and Muthian Sivathanu. 2020. Unsupervised clustering using pseudo-semisupervised learning.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018a. A deep generative framework for paraphrase generation. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018b. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

Masato Ishii. 2021. Semi-supervised learning by selective training with pseudo labels via confidence estimation. *ArXiv*, abs/2103.08193.

Heinrich Jiang and Ofir Nachum. 2020. Identifying and correcting label bias in machine learning. In *AISTATS*.

Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. 2012. Undoing the damage of dataset bias. In *ECCV*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Yi Li and Nuno Vasconcelos. 2019. Repair: Removing representation bias by dataset resampling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9564–9573.

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data boost: Text data augmentation through reinforcement learning guided conditional generation. *arXiv preprint arXiv:2012.02952*.

Louis Mahon and Thomas Lukasiewicz. 2021. Selective pseudo-label clustering. *ArXiv*, abs/2107.10692.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Yassine Ouali, Céline Hudelot, and Myriam Tami. 2020. An overview of deep semi-supervised learning. *ArXiv*, abs/2006.05278.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, Brussels, Belgium. Association for Computational Linguistics.

Tengyu Ma Sanjeev Arora, Yingyu Liang. 2017. A simple but tough-to-beat baseline for sentence embeddings.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2021. Can question generation debias question answering models? a case study on question–context lexical overlap. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 63–72, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Katherine Stasaski, Grace Hui Yang, and Marti A. Hearst. 2020. More diverse dialogue datasets via diversity-informed data collection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4958–4968, Online. Association for Computational Linguistics.

Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. 2015. A deeper look at dataset bias. *ArXiv*, abs/1505.01257.

Isaac Triguero, Salvador García, and Francisco Herrera. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2):245–284.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9.

Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Maximilian Wich, Hala Al Kuwatly, and Georg Groh. 2020. Investigating annotator bias with a graph-based approach. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 191–199, Online. Association for Computational Linguistics.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong. Association for Computational Linguistics.

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

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Online. Association for Computational Linguistics.

Xiang Zhou, Heba Elfardy, Christos Christodoulopoulos, Thomas Butler, and Mohit Bansal. 2021. Hidden biases in unreliable news detection datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2482–2492, Online. Association for Computational Linguistics.

Xiaojin Jerry Zhu. 2005. Semi-supervised learning literature survey.

SSL	Search for the George and the Big Bang TV show this current book is worth five I want to go to the Freight House in Gabon Give four points to Leven Thumps and the Gateway to Foo Find the trailer for Seven Year Itch.
Clustering	Find a TV show called Union. I'm looking for the song called Standing for Something. Please look up The Immortals television show. Please get me The National Medical Journal of India game. Find Half Cut Tea.
EDA	show the put yourself in his berth place game show the inwards put yourself in his place game his the put yourself in show place game show the put yourself game his place in show the put yourself in his place gimpy
Paraphrasing	Find me the trailer for The Incredible Hulk Find me the trailer for The Matrix How can I get a copy of the book The Art of Playing the Game Where can I find the trailer for The Man Who Fell to Earth How can I watch the movie The Secret Garden
In-context Learning	Add Put Yourself in His Place to Wish List Add Put Yourself in His Place to Wishlist Add the game Put Yourself in His Place Add the game Put Yourself in His Place to your Web browser. Add the game Put Yourself in His Place to your Web site.

Table 5: Examples of labeled data generated through various data augmentation methods.

A Example Appendix

Setting	ATIS	TOP	SNIPS
Full data	97.94%	94.16%	98.86%
Random down-sampling, 10% data			
Random	88.58%	98.08%	91.69%
Class dependent bias injection, 10% data			
($K = 1$ close to centroid)	83.68%	82.85%	92.35%
($K = 1$ away from centroid)	87.70%	82.95%	92.85%
($K > 1$ away from centroid)	89.25%	87.16%	93.92%
($K > 1$)	89.53%	87.64%	94.28%
Class independent bias injection, 10% data			
($K > 1$)	85.55%	89.30%	94.12%

Table 6: Baseline results, trained with 10% labelled data

Setting	ATIS	TOP	SNIPS
Random down-sampling, 5% data			
Random	85.81%	90.43%	96.08%
Class dependent bias injection, 5% data			
($K = 1$ close to centroid)	80.49%	80.47%	90.30%
($K = 1$ away from centroid)	81.47%	79.15%	89.40%
($K > 1$ away from centroid)	86.49%	84.93%	90.44%
($K > 1$)	86.00%	83.82%	89.61%
Class independent bias injection, 5% data			
($K > 1$)	80.84%	85.88%	76.80%

Table 7: Baseline results, trained with 5% labelled data

B Results with 5% and 10% of datasets

Dataset: ATIS								
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B		
Random down-sampling	88.6%	-0.365%	3.03%	5.3%	5.23%	5.42%	5.21%	3.01%
Class dependent bias injection:								
($K = 1$ close to centroid)	83.7%	0.205%	-1.37%	3.36%	4.92%	5.57%	5.54%	1.04%
($K = 1$ away from centroid)	87.7%	-0.822%	-6.36%	-0.308%	1.82%	1.87%	2.13%	-5.74%
($K > 1$ away from centroid)	89.1%	-0.0571%	0.0685%	2.49%	1.44%	2.53%	3.7%	-0.331%
($K > 1$)	89.3%	1.47%	0.753%	2.68%	2.29%	3.61%	3.47%	1.72%
Class independent bias injection:								
($K > 1$)	85.6%	0.0114%	4.77%	4.93%	6.08%	6.78%	6.6%	3.48%
Dataset: Top								
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B		
Random down-sampling	91.7%	0.144%	0.0155%	0.772%	-1.41%	-1.9%	-1.97%	0.00221%
Class dependent bias injection:								
($K = 1$ close to centroid)	82.9%	0.365%	-2.62%	3.67%	-0.822%	0.653%	2.1%	2.64%
($K = 1$ away from centroid)	83%	0.408%	-3.38%	3.38%	-0.763%	-0.487%	1.38%	1.36%
($K > 1$ away from centroid)	86.9%	0.449%	-2.21%	0.845%	-2.94%	-2.1%	-0.113%	0.332%
($K > 1$)	86.6%	0.718%	-2%	1.16%	-2.87%	-2.04%	-0.481%	0.426%
Class independent bias injection:								
($K > 1$)	89.3%	0.177%	0.195%	1.11%	-0.323%	-0.672%	-1.58%	1.03%
Dataset: Snips								
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B		
Random down-sampling	98.1%	0.0857%	-0.0714%	0.329%	0.214%	0.2%	0.214%	-0.171%
Class dependent bias injection:								
($K = 1$ close to centroid)	92.4%	1.06%	2.7%	4.4%	3.94%	4.84%	4.91%	3.63%
($K = 1$ away from centroid)	92.9%	0.829%	1.43%	3.21%	3.11%	4%	3.74%	2.69%
($K > 1$ away from centroid)	94.6%	0.0857%	1.5%	2.51%	1.94%	2.44%	2.56%	1.97%
($K > 1$)	94.6%	0.557%	1.27%	2.63%	2.17%	2.59%	2.54%	1.66%
Class independent bias injection:								
($K > 1$)	94.1%	0.257%	2.27%	3.1%	2.69%	3.1%	3.04%	2.71%

Table 8: Relative improvement over the baseline model, trained with 10% labelled data

Dataset: ATIS								
	Baseline	SSL	Clustering	EDA	Gen_20Bp5	Gen_20Bp7	Gen_20Bp9	Gen_5B
Random down-sampling	85.8%	-0.525%	4.1%	3.54%	4.04%	5.76%	5.9%	2.51%
Class dependent bias injection:								
($K = 1$ close to centroid)	80.5%	-0.297%	-5.29%	3.82%	7.51%	7.68%	8.06%	1.88%
($K = 1$ away from centroid)	81.5%	0.103%	-16.5%	1.4%	5.87%	7.47%	7.65%	-2.17%
($K > 1$ away from centroid)	84.9%	2.07%	-4.46%	2.42%	3.39%	4.84%	5.47%	1.23%
($K > 1$)	87.2%	-1.4%	-5.32%	1.06%	1.77%	1.54%	2.13%	-0.982%
Class independent bias injection:								
($K > 1$)	80.8%	-0.0685%	6.37%	6.53%	7.97%	9.35%	9.47%	5.76%
Dataset: Top								
	Baseline	SSL	Clustering	EDA	Gen_20Bp5	Gen_20Bp7	Gen_20Bp9	Gen_5B
Random down-sampling	90.4%	0.188%	-0.0177%	1.06%	-1.72%	-2.21%	-2.04%	0.0122%
Class dependent bias injection:								
($K = 1$ close to centroid)	80.5%	0.358%	-6.9%	1.75%	-3.77%	-3.24%	0.094%	0.421%
($K = 1$ away from centroid)	79.2%	0.374%	-6.7%	2.84%	-1.6%	-1.01%	0.811%	1.39%
($K > 1$ away from centroid)	84.7%	0.661%	-4.24%	0.841%	-3.95%	-2.24%	-0.583%	0.0343%
($K > 1$)	82.7%	0.679%	-6.14%	0.543%	-2.28%	-1.74%	1.24%	0.677%
Class independent bias injection:								
($K > 1$)	85.9%	0.222%	-0.885%	1.68%	-0.5%	-0.0221%	-1.11%	1.66%
Dataset: Snips								
	Baseline	SSL	Clustering	EDA	Gen_20Bp5	Gen_20Bp7	Gen_20Bp9	Gen_5B
Random down-sampling	96.1%	0.171%	1.2%	2.14%	2.07%	1.89%	1.97%	1.64%
Class dependent bias injection:								
($K = 1$ close to centroid)	90.3%	1.04%	3%	4.1%	5.04%	5.99%	6.27%	4.66%
($K = 1$ away from centroid)	89.4%	0.957%	2.73%	4.74%	4.53%	6.26%	6.54%	5.21%
($K > 1$ away from centroid)	89.9%	1.84%	3.71%	5.34%	5.47%	5.81%	6.51%	5.31%
($K > 1$)	90.2%	2.47%	4.06%	5.13%	4.73%	5.84%	5.93%	5.5%
Class independent bias injection:								
($K > 1$)	76.8%	0.371%	14.9%	12.5%	16.1%	17.3%	16.4%	14.9%

Table 9: Relative improvement over the baseline model, trained with 5% labelled data

Dataset: ATIS						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	0.00204	0.00189	0.000155	0.000148	0.000121	1.75e-05
Class dependent bias injection:						
($K = 1$ close to centroid)	0.000798	0.000999	0.00452	0.00147	0.000381	0.00284
($K = 1$ away from centroid)	0.000198	0.000245	0.0178	0.000325	0.000185	0.0055
($K > 1$ away from centroid)	0.000177	0.000323	0.000737	0.000547	0.000269	0.000595
($K > 1$)	0.000121	0.000219	0.000279	0.000284	0.000438	0.000347
Class independent bias injection:						
($K > 1$)	0.00115	0.00111	0.000313	0.000267	0.000179	0.000319
Dataset: Top						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	1.23e-05	1.58e-05	1.16e-05	6.18e-07	8.12e-06	1.75e-06
Class dependent bias injection:						
($K = 1$ close to centroid)	0.00054	0.000612	0.000738	0.000157	0.000333	8.69e-05
($K = 1$ away from centroid)	0.000602	0.000585	0.00143	0.000498	0.000832	0.000813
($K > 1$ away from centroid)	0.000151	0.000309	0.000732	0.000168	0.000252	8.43e-05
($K > 1$)	0.00013	0.000219	0.000303	4.03e-05	0.00016	0.000141
Class independent bias injection:						
($K > 1$)	8.13e-05	8e-05	5.1e-05	2.91e-05	2.95e-05	1.38e-05
Dataset: Snips						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	2.42e-05	2.28e-05	6.71e-06	3.04e-06	4.67e-06	4.57e-06
Class dependent bias injection:						
($K = 1$ close to centroid)	0.00206	0.00119	0.000432	8.57e-05	0.000291	2.02e-05
($K = 1$ away from centroid)	0.000584	0.000758	0.000345	0.000228	0.000262	0.000209
($K > 1$ away from centroid)	0.000123	0.000166	9.23e-05	3.19e-05	7.8e-05	2.64e-05
($K > 1$)	0.000580	0.000173	0.00014	4.61e-05	4.74e-05	5.57e-05
Class independent bias injection:						
($K > 1$)	0.00262	0.0025	0.000324	6.99e-05	7.23e-05	1.81e-05

Table 10: Variance of results over 10 different runs, trained with 10% labelled data

Dataset: ATIS						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	0.000714	0.000572	0.000259	2.37e-05	2.72e-05	3.02e-05
Class dependent bias injection:						
($K = 1$ close to centroid)	0.00128	0.00107	0.00565	0.000599	0.000519	0.0043
($K = 1$ away from centroid)	0.00175	0.00137	0.0285	0.000691	0.000185	0.00654
($K > 1$ away from centroid)	0.000832	0.000754	0.003	0.000785	0.000735	0.000596
($K > 1$)	0.000432	0.000505	0.000845	0.000529	0.00051	0.00184
Class independent bias injection:						
($K > 1$)	0.0022	0.00221	0.000829	0.000176	0.000281	0.000309
Dataset: Top						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	1.48e-05	1.34e-05	2.19e-05	2.42e-06	1.62e-05	4.26e-06
Class dependent bias injection:						
($K = 1$ close to centroid)	0.000827	0.000816	0.00184	0.000347	0.00116	0.00047
($K = 1$ away from centroid)	0.00148	0.00144	0.00555	0.00098	0.0013	0.00257
($K > 1$ away from centroid)	0.000427	0.000393	0.00127	0.000423	0.000659	0.000508
($K > 1$)	0.000132	0.000489	0.00078	0.000559	0.00033	0.000393
Class independent bias injection:						
($K > 1$)	8.28e-05	9.42e-05	0.000149	4.15e-05	0.000101	3.48e-05
Dataset: Snips						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	0.000536	0.00046	6.37e-05	4.98e-06	9.16e-06	1.12e-05
Class dependent bias injection:						
($K = 1$ close to centroid)	0.000768	0.000614	0.000581	0.000499	0.000529	0.000116
($K = 1$ away from centroid)	0.00104	0.000788	0.00106	0.000408	0.000626	0.000257
($K > 1$ away from centroid)	0.000124	0.000918	0.000385	0.000104	0.000267	6.91e-05
($K > 1$)	0.00524	0.000203	0.00019	0.000143	0.000318	7.71e-05
Class independent bias injection:						
($K > 1$)	0.0153	0.0149	0.00232	0.0047	0.00165	0.00355

Table 11: Variance of results over 10 different runs, trained with 5% labelled data

Dataset: ATIS						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	0.0314	0.0249	0.00123	0.000183	2.58e-05	8.35e-05
Class dependent bias injection:						
($K = 1$ close to centroid)	0.0117	0.0125	0.0204	0.000693	0.00622	0.00349
($K = 1$ away from centroid)	0.00103	0.000464	0.0517	0.000679	0.00107	0.00343
($K > 1$ away from centroid)	0.000176	0.00122	0.0159	0.000216	0.000358	0.000931
($K > 1$)	0.000163	0.00645	0.00957	0.00646	0.00469	0.00498
Class independent bias injection:						
($K > 1$)	0.00163	0.00114	0.00569	0.000225	0.000391	0.000617
Dataset: Top						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	0.000299	0.000316	5e-05	7.41e-06	4.7e-05	1.52e-05
Class dependent bias injection:						
($K = 1$ close to centroid)	0.000916	0.000904	0.00683	0.000542	0.00123	0.000852
($K = 1$ away from centroid)	0.00146	0.00138	0.00914	0.00214	0.00134	0.00312
($K > 1$ away from centroid)	0.000116	0.00148	0.00203	0.000689	0.000759	0.00107
($K > 1$)	0.000126	0.000963	0.00262	0.00117	0.000622	0.000568
Class independent bias injection:						
($K > 1$)	0.00165	0.00157	0.00171	0.00254	0.000505	0.00163
Dataset: Snips						
	Baseline	SSL	Clustering	EDA	Gen_20B	Gen_5B
Random down-sampling	0.00187	0.00137	0.000105	0.000651	0.000222	5.56e-05
Class dependent bias injection:						
($K = 1$ close to centroid)	0.00469	0.00393	0.00171	0.00199	0.00869	0.000947
($K = 1$ away from centroid)	0.00403	0.003	0.00141	0.00276	0.00336	0.000746
($K > 1$ away from centroid)	0.000576	0.00549	0.00172	0.000786	0.00283	0.000714
($K > 1$)	0.000271	0.00473	0.00175	0.00191	0.000665	0.000594
Class independent bias injection:						
($K > 1$)	0.0172	0.0172	0.011	0.0272	0.0187	0.0174

Table 12: Variance of results over 10 different runs, trained with 1% labelled data