Sampling bias in NLU models: Impact and Mitigation

Zefei Li, Anil Ramakrishna, Anna Rumshisky, Andy Rosenbaum, Saleh Soltan, Rahul Gupta

Amazon Alexa

{lzefei, aniramak, arrumshi, andros, ssoltan, gupra}@amazon.com

Abstract

Natural Language Understanding (NLU) systems such as chatbots or virtual assistants have seen a significant rise in popularity in recent times, thanks to availability of large volumes of user data. However, typical user data collected for training such models may suffer from sampling biases due to a variety of factors. In this paper, we study the impact of bias in the training data for intent classification task, a core component of NLU systems. We experiment with three kinds of data bias settings: (i) random down-sampling, (ii) class-dependent bias, and (iii) class-independent bias injection. For each setting, we report the loss in model performance and survey strategies to mitigate the loss from two families of methods: (i) semi-supervised learning (SSL), and (ii) synthetic data generation. Overall, we find that while both methods perform well with random down-sampling, synthetic data generation out-performs SSL when only biased training data is available.

1. Introduction

Data collection is an integral part of training any Machine Learning (ML) system and the data collection protocol can significantly impact the performance of the ML model. While access to an arguably unrestricted data source for unbiased data collection in large volumes is desirable, it may not always be feasible. For instance, under certain conditions, data collection protocols may dictate separate data collection per label of interest: while building a commercial Natural Language understanding (NLU) model, the developer may start by sourcing data from a particular group of end users or annotators to generate requests related to playing music first, and a later time source data from a separate group for a new feature (such as querying for weather). This change in user population can induce subtle changes in the data subsets corresponding to the two intent labels *PlayMusic* and *GetWeather*.

Another unexpected source of bias is user privacy, due to which gathering labeled data in large volumes becomes challenging leading to data collection being restricted to a biased sub-sample of users. For example, while building the NLU system, due to privacy concerns the developers are sometimes constrained to explicitly get user approval to have their data stored/labeled for downstream model use; in this scenario, it is likely that only a small section of user population would donate their data, leading to bias (say, in intent distributions) in the retained training dataset.

In this work, we study the impact of such biases in the intent classification sub-component of natural language understanding, introduced during the dataset collection process, and survey the efficacy of a number of mitigation strategies. We simulate settings that mimic different kinds of biases that can be introduced during data collection. We start with uniformly random down-sampling, and subsequently introduce biases under data collection protocols that either collect data for supported labels independently or 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 there. In addition, low availability of data is an increasingly realistic scenario in building industrial ML systems given emerging privacy considerations [1]. Furthermore, we survey the benefits of a variety of techniques from two broad data augmentation strategies: (i) semi-supervised learning assuming availability of unlabeled data, and (ii) synthetic data generation. We assess their impact in recovering degradations in model performance arising from the low volume and biased training data.

2. Related Work

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 [1]. 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 [2, 3], the design of the annotation guidelines and prompts [4], the subjective judgements made by individual annotators [5] and, the decisions about how to split a dataset into training, validation, and test sets [2]. Models trained on these biased datasets may then learn to exploit dataset-specific artifacts [6, 7], 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 [8–10], data augmentation methods [11, 12], and bias-aware learning algorithms [13–15]. However, few prior studies have examined this problem in the NLU domain. In this work, we address this gap by surveying a number of methods to create biased subsamples in existing, publicly available datasets. We use these methods to 1) create several benchmark text classification datasets with different types of bias and, 2) evaluate the performance of a variety of techniques to mitigate these biases.

3. Bias simulations

Depending on the underlying factors involved in the dataset collection scenario, a variety of biases may creep into the obtained data. We discuss three such bias conditions below, with illustrations shown in Figure 1.

3.1. Random down-sampling

In this scenario, we assume availability of data from the real world distribution. This scenario is likely, for example, when an ML practitioner has access to the process governing data generation, but they are constrained to sample a small portion of the data. We randomly downsample our available datasets to a fraction of its original size to simulate this scenario. This method is expected to provide a smaller number of datapoints, but may not introduce any bias in the sampled data.

3.2. Class-dependent bias injection

In many applications, practitioners are constrained to gather data per class. For example, in an industrial setting, one may launch ML models with a pre-defined class support (e.g. an intent classification model that classifies 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.

In our to simulate this scenario, given a class, we obtain K seed datapoints from amongst the datapoints belonging to that class. Given these seed datapoints, we select utterances near them (where distance is defined on an appropriate 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 nearby utterances can be 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 computed based on all the available datapoins for the class at hand, as defined on the chosen embedding space.

3.3. Class-independent bias injection

In this scenario, the practitioner first collects data for the pre-defined class support, gets them annotated 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. To inject such a bias, 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.

4. Experiments

4.1. Datasets

We use three intent classification datasets for our experiments: i) The ATIS Intent Classification Dataset [16] dataset, ii) The Semantic Parsing for Task Oriented Dialog using Hierarchical Representations (TOP) [17] dataset, iii) The SNIPS Natural Language Understanding benchmark [18]. For each of these datasets, we created biased subsets using the sampling scenarios described in Section 3, and train intent classifiers on each of these sets. We further experiment with different degrees of data reduction in each of our biased sampling scenarios, with the constraint that at least one data point is available per class in each simulation. This is important as unconstrained severe under-sampling may lead to a reduced class support, as datapoints from some classes may not be sampled. In our experiments, we reduced the data size to to 1%/5%/10%, of its original volume and report results on each of these. We continue selecting nearest utterances to the selected seed utterances until we reach the target proportion.

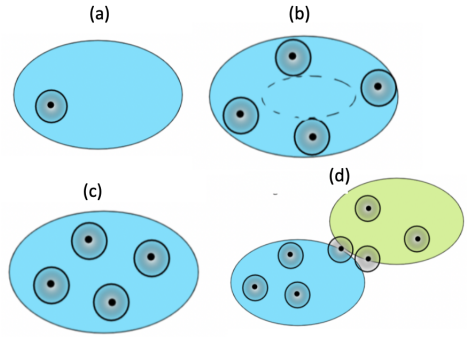

Figure 1: *Given that data in a chosen class is shown using the blue ellipse, (a) shows sampling with a single seed ($K = 1$) with the seed selected away from the class centroid, (b) shows sampling with multiple seeds ($K > 1$) with seeds away from centroid, (c) shows sampling with several randomly selected seeds, and (d) shows sampling with seeds selected randomly irrespective of the class (green ellipse denotes a class different from the blue one).*

4.2. Implementation details

For each scenario described in Section 3, we operate in an utterance embedding space based on the smooth inverse frequency (SIF) method [19]. SIF has been shown to be a strong, yet simple method to obtain sentence embeddings. We select seed utterances in the SIF embeddings space and select proximal utterances based on L2 norm. We also note that in the real world the process for biased data generation is unlikely to be available. Therefore, we do not use SIF based embeddings in any of our subsequent mitigation strategies to counter the effect of the biased data samples. We fine-tune a BERT base model (with $\sim$110M parameters) on the available labeled data for all our classification tasks. We create 10 versions of datasets under each setting and present average performance across them. Results for our baselines are shown in Table 1.

4.3. Observations

1. **All of the down-sampling regimes lead to some form of performance degradation**, suggesting the need for suitable mitigation strategies.
2. **While the random down-sampling setting had the least degradation in TOP and SNIPS, it degrades the most in ATIS.** We expected random down-sampling to show the least degradation compared to the full data baseline, since it preserves class distributions across data samples. However, this is not the case in ATIS dataset, especially in the setting sampled down to 1% of its size. This is likely due to the considerably smaller size of this dataset, where severe under-sampling (to 1%) leaves room for just 1-2 samples per class, as shown in Table 2. While this result is expected, it has important implications for the few-shot learning scenarios, in which sampling data to match the true distribution becomes impossible. Such scenarios are common in

Table 1: *Baseline results, trained with 1%/5%/10% of labelled data*

Dataset	ATIS	TOP	SNIPS	ATIS	TOP	SNIPS	ATIS	TOP	SNIPS
Full data	97.94	94.16	98.86		-			-	
Data proportion		1%			5%			10%	
Random down-sampling	66.52	83.50	85.81	85.81	90.43	90.08	88.58	98.08	96.69
Class dependent bias injection									
($K = 1$ close to centroid)	70.59	73.45	68.51	80.49	80.47	90.30	83.68	82.85	92.35
($K = 1$ away from centroid)	72.30	72.22	75.22	81.47	79.15	89.40	87.70	82.95	92.85
($K > 1$ away from centroid)	80.77	77.65	80.77	86.49	84.93	90.44	89.25	87.16	93.92
($K > 1$ randomly chosen)	73.69	74.39	75.04	86.00	83.82	89.61	89.53	87.64	94.28
Class independent bias injection									
($K > 1$)	72.21	72.76	34.40	80.84	85.88	76.80	85.55	89.30	94.12

Table 2: *Number of utts. selected per intent in ATIS*

Intent/Ratio	10%	1%	10%	1%
Intent/Ratio	random sampling		Class independent bias injection	
Intent/Ratio				
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

Table 3: *Number of utts. in each intent of SNIPS with class independent biased sampling*

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

real-world uses of NLU systems, where very limited data may be available when new classes (new intents) are introduced. In particular, gathering biased data per-class yields more samples for under-represented classes (e.g. capacity/distance), leading to better accuracy in a few-shot setting.

3. **($K > 1$ 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 to gather valuable annotated data.

4. **The class-independent bias injection setting ($K > 1$) shows severe under-performance for SNIPS.** We observe an average performance (over 10 runs) of 34.4% in the stated setting in SNIPS when dealing with 1% data, but the performance is considerably better when using 5% or 10% of the data. To further examine this, we list the number of datapoints per class from the class independent bias injection experiment in Table 3 (sampled from one of the 10 runs). From this table, we can see that severe under-sampling in SNIPS leads to a skew in the training data with intents like *GetWeather* and *SearchScreeningEvent* observing far fewer datapoints compared to the 10% and 5% experiments. We observe that these intents, while frequent in the overall population, are tightly clustered in the embedding space and if a seed is not chosen close to their 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.

5. Mitigation Strategies

To mitigate the performance degradation observed in the baseline experiments earlier, we explored two broad categories of *data augmentation*, as described below.

5.1. Semi-Supervised Learning (SSL)

In SSL, we assume availability of a large volume unlabeled datapoints, which is frequently the case for many real world applications. We describe two *pseudo-labeling* strategies below.

Self-learning based SSL: in this method, we train a seed model on the available labeled data and pseudo-label the unlabeled data with the seed model. For both seed and augmented models, we use a BERT-based pre-trained model trained from ConSERT [20] and fine-tune it on the labeled data.

Clustering-based SSL: in this approach we propagate labels from the labeled datapoints to neighboring un-labeled datapoints. Unlabeled utterances help learn the underlying cluster distributions of the data while the labeled utterances assign labels to these clusters. Similar to [21], we use the pre-trained language model BERT to produce sentence embeddings for both labeled and unlabeled datapoints. We use K-means clustering with the number of clusters set to the number of known classes [22]. We expect that each cluster represents a set of semantically similar sentences. To ensure quality of the generated labels, we only select the most confident clusters in our experiments and label these using the labeled datapoints present in each cluster. Following [23], we only retain "pure" clusters, defined as those in which: (a) at least 1% of the datapoints in a given cluster need to be labeled, and (b) the majority class amongst the labeled datapoints needs to account for at least 80% of the labeled datapoints. Given such clusters, all unlabeled datapoints are assigned the majority class and are then augmented to the labeled dataset for subsequent training.

5.2. Synthetic Data Generation

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

Easy Data Augmentation (EDA) [24] is a data augmentation technique which uses strategies such as synonym replacement, random synonym insertion, random swap of two words and random word deletion to synthesize new training examples. It creates 9 new synthetic utterances for each labeled utterance using these techniques. While the heuristic behind EDA is simple, it has shown to outperform several data generation baselines.

Back Translation (BT) [25] in BT, a machine translation (MT) system is applied to translate text from the source lan-

Table 4: *Accuracy of models, trained with 1% of labelled data and augmented data from each method*

Dataset: ATIS								
Full data baseline	97.94							
	Baseline	SSL	Clustering	EDA	ICL_p5	ICL_p7	ICL_p9	BT
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
($K > 1$ randomly chosen)	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	ICL_p5	ICL_p7	ICL_p9	BT
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
($K > 1$ randomly chosen)	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	ICL_p5	ICL_p7	ICL_p9	BT
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
($K > 1$ randomly chosen)	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

guage to a target pivot language, then back again. By sampling from the N-best hypotheses in both directions, BT can produce a large number of paraphrases. We fine-tune a 5B parameter seq2seq model [26] on WMT 2014 data [27], using a single model for en→fr and fr→en, with instruction prompts to control both language directions: "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 was 41 for ATIS, 51 for SNIPS, and 36 for TOP.

In-Context Learning (ICL): We use a 20B parameter language model [28] to generate new data using a small sample of labeled data from the task at hand as context. In our experiments, for each intent in each dataset, we fine-tune the language model using 3 randomly sampled exemplar utterances using the template *Example from [intent_name] intent: utterance_text*. For example, for the flight intent in ATIS dataset, we would fine-tune using utterances of the form *Example with [flight] intent: do you have an early morning direct flight from philadelphia to pittsburgh?*. Following this procedure, we generate 27 samples of the same intent by letting the model continue token generation after the prompt (for example *Example with [flight] intent:*). For data generation, we use nucleus sampling [29] with $p = 0.5, 0.7, 0.9$, denoted as ICL_p5, ICL_p7 and ICL_p9, respectively.

5.3. Discussion

In our mitigation experiments, we focus on the baseline with 1% training data since this is the most challenging setup with most degradations. For SSL, we use data disjoint from the biased sampling set for the unlabeled data. We use the same BERT architecture from the baseline experiments for fine-tuning on the augmented datasets. Table 4 summarizes our results.

We observe that the synthetic data obtained via generative models trained 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 case of low labeled data setting, the seed set of labels may not provide a high quality starting point. Even simple data augmentation methods such as EDA beat semi-supervised learning based methods. In absence of a diverse and representative labeled datapoints, pseudo-labeling unlabelled data can be challenging. Secondly, as reported in other literature, large models perform the best in yielding highest quality data. In particular, in context learning tuned with a handful of data beats other methods in most settings. This is consistent with results reported elsewhere [30].

6. Conclusion

In many real-world ML settings, data collection may be biased due to various reasons. In this paper, we simulate several types of sampling biases in an intent classification system, motivated by real-world scenarios. We observed models trained on biased samples outperforming models trained on random sub-sampled data, since under-represented classes did not get severely down-sampled. To mitigate such biases, we test two sets of data augmentation methods that make use of in-domain unlabeled data and data generation models trained using large volumes of natural language corpora, and observe stronger performance in models trained on data augmented with synthetic data (compared with pseudo-labeled in-domain data). In the future, we aim to extend this analysis to other biases (e.g., when data is missing for some classes) and tasks in NLU.

7. References

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

[2] Xiang Zhou, Heba Elfardy, Christos Christodoulopoulos, Thomas Butler, and Mohit Bansal, "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*, Online, Apr. 2021, pp. 2482–2492, Association for Computational Linguistics.

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

[4] Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith, "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)*, Vancouver, Canada, Aug. 2017, pp. 15–25, Association for Computational Linguistics.

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

[6] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith, "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)*, New Orleans, Louisiana, June 2018, pp. 107–112, Association for Computational Linguistics.

[7] Masatoshi Tsuchiya, "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, May 2018, European Language Resources Association (ELRA).

[8] Mor Geva, Yoav Goldberg, and Jonathan Berant, "Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets," *arXiv preprint arXiv:1908.07898*, 2019.

[9] Roy Schwartz, Maarten Sap, Ioannis Konstas, Li Zilles, Yejin Choi, and Noah A Smith, "The effect of different writing tasks on linguistic style: A case study of the roc story cloze task," *arXiv preprint arXiv:1702.01841*, 2017.

[10] Maximilian Wich, Hala Al Kuwatly, and Georg Groh, "Investigating annotator bias with a graph-based approach," in *Proceedings of the fourth workshop on online abuse and harms*, 2020, pp. 191–199.

[11] Xiang Zhou and Mohit Bansal, "Towards robustifying NLI models against lexical dataset biases," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020, pp. 8759–8771, Association for Computational Linguistics.

[12] Ji Ho Park, Jamin Shin, and Pascale Fung, "Reducing gender bias in abusive language detection," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct.-Nov. 2018, pp. 2799–2804, Association for Computational Linguistics.

[13] Heinrich Jiang and Ofir Nachum, "Identifying and correcting label bias in machine learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 702–712.

[14] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer, "Learning to model and ignore dataset bias with mixed capacity ensembles," *arXiv preprint arXiv:2011.03856*, 2020.

[15] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba, "Undoing the damage of dataset bias," in *European Conference on Computer Vision*. Springer, 2012, pp. 158–171.

[16] Charles T. Hemphill, John J. Godfrey, and George R. Doddington, "The ATIS spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990.

[17] Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis, "Semantic parsing for task oriented dialog using hierarchical representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct.-Nov. 2018, pp. 2787–2792, Association for Computational Linguistics.

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

[19] Sanjeev Arora, Yingyu Liang, and Tengyu Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *International conference on learning representations*, 2017.

[20] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," in *ACL*, 2021.

[21] Roee Aharoni and Yoav Goldberg, "Unsupervised domain clusters in pretrained language models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020, pp. 7747–7763, Association for Computational Linguistics.

[22] Louis Mahon and Thomas Lukasiewicz, "Selective pseudo-label clustering," *ArXiv*, vol. abs/2107.10692, 2021.

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

[24] Jason Wei and Kai Zou, "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)*, Hong Kong, China, Nov. 2019, pp. 6382–6388, Association for Computational Linguistics.

[25] Rico Sennrich, Barry Haddow, and Alexandra Birch, "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)*, Berlin, Germany, Aug. 2016, pp. 86–96, Association for Computational Linguistics.

[26] Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese, "Linguist: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging," *arXiv preprint arXiv:2209.09900*, 2022.

[27] 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, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, June 2014, pp. 12–58, Association for Computational Linguistics.

[28] Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al., "Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model," *arXiv preprint arXiv:2208.01448*, 2022.

[29] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[30] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen, "What makes good in-context examples for gpt-3?," *arXiv preprint arXiv:2101.06804*, 2021.