De-biasing training data distribution using targeted data enrichment techniques

Dieu-Thu Le
Amazon Alexa AI
Germany
deule@amazon.com

Jose Garrido Ramas
Amazon Alexa AI
Germany
jrramas@amazon.com

Yulia Grishina
Amazon Alexa AI
Germany
yuliag@amazon.com

Kay Rottmann
Amazon Alexa AI
Germany
krrottm@amazon.com

ABSTRACT

In this paper, we introduce a targeted data enrichment framework to mitigate the problem of biased training data distribution. In real world applications, it is often observed that the training data distribution differs from the online live traffic data due to multiple reasons such as topic changes, seasonalities, the nature of users. Our targeted data augmentation techniques generate samples that are most similar to those that are missing in the training data. The main idea behind the selection strategy is to fill in samples that are not yet well represented in the training data. Our framework consists of a semi-supervised learning (SSL) component and a synthetic data generation part. For SSL, we use a retrieval module with guided weights learned from a data drift model. We further discuss the problems of accumulated errors in SSL by introducing a low confidence SSL data selection strategy. For synthetic data augmentation, we use masked language model data generation by using a concept of word *replaceability* to produce meaningful samples. We report our results on two large commercial datasets in real world applications and show that our framework could improve the error rates in almost all domains, and on average up to 4.6%. We also report the results of the data augmentation techniques on two public datasets, where we see improvements in both cases.

KEYWORDS

data augmentation, semi-supervised learning, text classification, biased datasets, synthetic data generation

1 INTRODUCTION

In natural language understanding (NLU) applications, it is often observed that the training data distribution might differ from the online live data, in which the application is being deployed. This can be due to multiple reasons, such as the changes of topics of interest, the nature of the users, the seasonality patterns as well as other general content drift. Intuitively, the prediction of the new data is often based on the assumption that the training is representative of the online data. This assumption is however often violated leading to degradation in online performance and the deployed models are not working well as expected in practice.

Data enrichment techniques that either make use of unlabelled data such as semi-supervised learning (SSL) or synthetic data augmentation [Kumar et al. 2019; Sennrich et al. 2016; Xie et al. 2019] have recently become a de-facto approach to deal with low regime data in machine learning problems. Recent studies [Berthelot et al. 2019; Cho et al. 2019; Tarvainen and Valpola 2017b] have shown the effectiveness of these approaches when the access to labelled data is limited and that it can significantly boost the performance of the systems with few to very limited numbers of training samples [Chapelle et al. 2009; Xie et al. 2019; Zhu and Goldberg 2009; Zhu 2005]. In this paper, instead of focusing on low data regimes, we show evidences that general data enrichment techniques are also particularly helpful when dealing with the problems of biased distributions in training data.

We introduce a framework for data enrichment to compensate the mismatch between the online data and the training data. This framework adapts based on the measurement of the difference in training data comparing to online data as a guidance, and the enriched data is generated using two techniques: semi-supervised learning and synthetic data generation to compensate for this difference.

Firstly, for SSL, we use semantic retrieval to search for relevant unlabelled data using heuristic filtering techniques: we use missing sets as guidance to search for relevant utterances, combining with out of distribution detection and filtering criterion. The idea behind self training is to minimize the entropy [Grandvalet and Bengio 2005] of the output distributions as a regularizer to drive the decision boundary from new learning examples. Furthermore, out of distribution detection and filtering techniques are applied to make sure that the added utterances are of value to the model.

Common SSL data selection usually limits to only those that the model is already confident with in order to deal with the problem of accumulated errors over time or sometimes called confirmation bias (i.e., samples that have been annotated correctly by the model will be added again to the model, at the same time, it might get overconfident on samples that it has annotated wrongly in earlier stages and magnify its errors) [Berthelot et al. 2019]. In order to deal with this problem, we further introduce a mechanism to find also low confidence samples, where we have evidences that they are correctly annotated using label propagation techniques.

Secondly, for synthetic data generation, we present a masked language model data generation strategy by calculating a masking probability for each token, which measures how *replaceable* each word is. The data generation process is also guided by taking into account the difference between online vs. training data distribution.

The main contributions and key findings of the paper are as follows.

- We introduce a general framework for data enrichment to mitigate the bias problem in training data. The main idea behind the enrichment framework is to measure the difference between training vs. online data using a data drift model and use it as a guided function for both SSL and synthetic data augmentation to generate *missing* data.
- We explain how debiased SSL works using a retrieval module to find which samples should be selected to mitigate the data distribution bias problem. We further discuss the well known problem of accumulated errors over time for self training model with SSL and introduce a mechanism how to safely enrich the training data with low confidence samples.
- We describe how we use masked language models for synthetic data augmentation by adopting the concept of word *replaceability* to generate more meaningful and proper samples to enrich the training data and overcome data bias.

We show through our experiments on two large commercial datasets of the spoken language understanding tasks how this framework helps to mitigate the common training data bias problem in practical applications, with an improvement of up to 4.6% error rate reduction. We also describe our lessons learned with technical details how to achieve the best performance through the ensembling filtering approach as well as the data drift model weighting techniques. We further report our experimental results on two public datasets using the data augmentation technique.

2 RELATED WORK

Semi-supervised learning is frequently utilised in NLP to augment low-traffic classes with additional data and overcome the class imbalance problem [Cho et al. 2019]. The goal of semi-supervised learning is to improve model's performance by leveraging unlabelled data. It has been proven to achieve high performance using fewer labeled data and a large amount of unlabelled data [Chapelle et al. 2009; Zhu and Goldberg 2009; Zhu 2005]. In the literature, many different approaches are described that define the selection and labelling of semi-supervised data. Since such unlabelled data is typically noisy, multiple state-of-the-art methods focused on overcoming that noise, such as loss correction [Li et al. 2020] or pseudo-labelling [Ortego et al. 2021]. Another approach widely used in NLP is the self-training approach [Zhou and Belkin 2014; Zhu 2005], which automatically annotates unlabelled training samples by the model trained on the available data. Afterwards, high confidence labels are selected and fed into the model, and the training is repeated. This way the model is empowered to use its own predictions to teach itself. However, one of the main drawbacks of these and other approaches is (i) their extensive use of high confidence labels (e.g., [Arazo et al. 2020; Sohn et al. 2020]) that may propagate errors if selected inaccurately and lead to the so-called confirmation bias [Liu and Tan 2021; Tarvainen and Valpola 2017a], and (ii) their inability to make use of the low-confidence labels that introduce additional data variation into the training data. In addition to model based labeling approaches, in recent years different approaches for model based synthetic data generation for creating new training data in natural language processing were introduced. [Feng et al. 2021] provides an overview over different approaches. In [Sennrich et al. 2016] a description of using back-translation for enriching machine translation based on monolingual data is given. In a similar direction is the idea of using paraphrases of input text, as for example described in [Kumar et al. 2019], to create new training samples via paraphrasing. Pretrained language model are another approach that gained a lot of attention recently. The idea is to use these pretrained language models to generate new examples accordingly. Examples for this can be found in [Kobayashi 2018] where randomly words are replaced by other words predicted by a pretrained bi-directional LSTM language model. Also on a word level is the idea of [Ng et al. 2020] operating. They use a transformer architecture as a denoising auto-encoder for reconstructing masked inputs showing strong performance improvements in applications.

Finally, data enrichment techniques always involve the risk of increasing the noise in the training data. Approaches to deal with noisy training data can focus on loss functions / training strategies that are robust to noise [Amid et al. 2019] [Ma et al. 2020], however if too much noise is present in the training data, and particularly if the noise is not random, eventually any model will learn the wrong interpretations. Filtering techniques, on the other hand, can filter out noise in augmented data before adding it to the training data, even in the extreme case that most of the augmented data has a wrong annotation. In [Li et al. 2019], a graph-filtering approach is used to select which unannotated data to add to the training. [Pleiss et al. 2020] proposes an "Area Under the Margin" metric, computed at different epochs, to identify mislabelled data. This metric will be high if, at the various epochs, the model considers an annotation has a high margin in the logits of the output layer. In [Laine and Aila 2016] SSL data is annotated using a temporal ensemble of models, which are trained a different amount of epochs and with different input augmentation conditions. We implement a filtering strategy based on this work, in which we only add data to training if there is agreement between a teacher model, trained during a high number of epochs, and a student model trained on less epochs and less data.

3 MITIGATING DATA BIAS WITH DATA ENRICHMENT TECHNIQUES

In this section, we describe our two data enrichment strategies using SSL and synthetic data generation with Masked Language Model (MLM) in a practical unbalanced data distribution scenario. Section 3.3 explains the debiased semi-supervised learning approach, section 3.4 describes how we use low confidence scored data selection for SSL. Section 3.5 introduces our techniques for synthetic data generation with MLM and section 3.6 explains how we use ensemble filtering to do post-filtering for all of our data selection before adding them to the final training data.

3.1 Problem statement

In natural language understanding (NLU) applications, it is common that the distribution of online data from users varies over time. For example, in smart home voice assistants, when there are releases of new devices with new features, customers requests will be different from those in the past. In this paper, we illustrate our framework

in an NLU application: recognizing domains, intents and named entities of a customer's utterance text.

Let $\mathcal{T}$ be the training data $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{T}$, where x_i refers to an utterance and y_i are the corresponding human annotated labels of the utterance text x_i.

Let $\mathcal{O} = \{(x_i, y_i)\}_{i=1}^{O}$ be a sample of the online data (i.e., unlabelled data) that represents the recent live traffic data from customers, which could change over time, e.g., under topic drift. y_i are the proxy labels of the utterance text x_i, which are obtained from an NLU model. The training data $\mathcal{T}$ and the online data $\mathcal{O}$ are drawn from two different distributions denoted as $P_{\mathcal{T}}$ and $P_{\mathcal{O}}$ respectively.

We define a data shift model S that measures the differences between $P_{\mathcal{T}}$ and $P_{\mathcal{O}}$ by projecting the training data and online data in the same embedding space, clustering the data and assign a weight for each data cluster. The cluster weight defines how much data in the training set is *missing* around this topic (i.e., how many utterances should be added to the training data to deal with the distribution shift). The intuition behind the data enrichment strategies is to measure the differences between $P_{\mathcal{T}}$ and $P_{\mathcal{O}}$, generate new training data to overcome the mismatch.

We define $S = \{(c_k, w_k)\}_{k=1}^{K}$ where c_k is the k cluster of utterances and w_k is the corresponding weight of the cluster k. Our aim is to find a set of unlabelled data $E = \{(x_i, y_i)\}_{i=1}^{E}$ to enrich the current $\mathcal{T}$ to minize the gap between the online and training data distribution: $\mathcal{T} = \mathcal{T} \cup E$.

Algorithm 1: Modeling data shift in the embedding space

Intput: • Set of utterances $U = \mathcal{T} \cup \mathcal{O}$, where $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{T}$, $\mathcal{O} = \{(x_i, y_i)\}_{i=1}^{O}$; $x_i \in \mathbb{R}^n$, n is the embedding size
• Number of cluster K
Output: Data shift model $S = \{(c_k, w_k)\}_{k=1}^{K}$
Initialize k cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly;
repeat
  for *every utterance* x_i *in* U **do**
    set cluster label $l^{(i)}$ to each utterance x_i:
    $l^{(i)} := \arg\min_j ||x_i - \mu_j||^2$
  end
  for *each* $k = 0, \ldots K$ **do**
    update centroids μ_k:
    $\mu_k := \frac{\sum_{k=1}^{U} 1\{l^{(i)}=k\} x_i\}}{\sum_{k=1}^{U} 1\{l^{(i)}=k\}}$
  end
until *convergence*;
for *each* $k = 0, \ldots K$ **do**
  Calculate weights for each cluster:

$$w_k = \frac{|x_i | l^{(i)} = k, x_i \in \mathcal{O}|}{|x_j | l^{(j)} = k, x_j \in \mathcal{T}|} \times \frac{|T|}{|O|} \quad (1)$$

end

3.2 Modeling data shift in embedding space

To measure the difference between $P_{\mathcal{T}}$ and $P_{\mathcal{O}}$, we compute the shift model $S = \{(c_k, w_k)\}_{i=1}^{K}$ as described in Algorithm 1.

Firstly, we project both training data and online data in a common embedding space. We use k-means to cluster the union set of these utterances to K clusters. The distance between each pair of utterances is calculated based on their L2 distance. In each cluster, we compute a *missing* weight, which estimates the difference between the proportion of online vs. training data in each cluster. Intuitively, the data enrichment framework aims at compensating this difference by generating new data through SSL as well as synthetic generation.

3.3 Debiased semi-supervised learning data selection

Semi-supervised learning has been used as a common method to alleviate the need of expensive labeled data. It has shown to be one of the most promising paradigms that leverage unlabeled data [Berthelot et al. 2019; Tarvainen and Valpola 2017b; Xie et al. 2019], especially in practical applications where the recent online traffic data gives the best estimation and representativeness of the customers' requests.

We develop a debiased semi-supervised learning (D-SSL) data acquisition strategy that selects live data resembling online data that is missing from the current training data. Intuitively, this will compensate for the missing annotated data in the training set and aid the learning process by providing additional data points that the model has not seen during training and preserve the distribution of the training data without being biased to only annotated utterances. The main steps of the D-SSL module is depicted in figure 1.

In particular, the selection of D-SSL data includes utterance semantic clustering and computation of weight w_k corresponding to the number of utterances that are missing in each cluster c_k. A retrieval module is employed to find relevant utterances for each cluster together with out of distribution detection (i.e., utterances that are irrelevant to the device functionalities). We employ filtering criterion such as NLU confidence score and ensemble model filtering to make sure that only utterances that are of value to the model are preserved and to avoid accumulating erroneous examples. The steps are explained in algorithm 2.

Let $E_{DSSL} = \{(x_i, y_i)\}_{i=1}^{E} | (x_i, y_i) \in \mathcal{U}$, where E_{DSSL} is the set of utterances that are selected from the D-SSL approach to enrich the original training data. From the set of unlabeled data $\mathcal{U}$, we find utterances that are most similar to those defined in S.

We used Sentence BERT encoder [Reimers and Gurevych 2019] with a pre-trained multilingual BERT and FAISS [Johnson et al. 2017] indexing to retrieve utterances in each cluster with the weights defined by the k-means clustering approach using L2 distance metrics. Typically, each utterance in the cluster is represented as vectors which are indexed and stored once. The search operation is performed on an index is the k-neareast-neighbor search, i.e., returning a matrix containing the IDs of the neighbors of the query vector, i.e., the average of all utterances in a cluster, sorted by their increasing distances.

Particularly, to retrieve most similar utterances in each cluster c_k, we use the average encoding of all utterances in each cluster and fetch w_k most similar utterances. Finally, we obtain the proxy

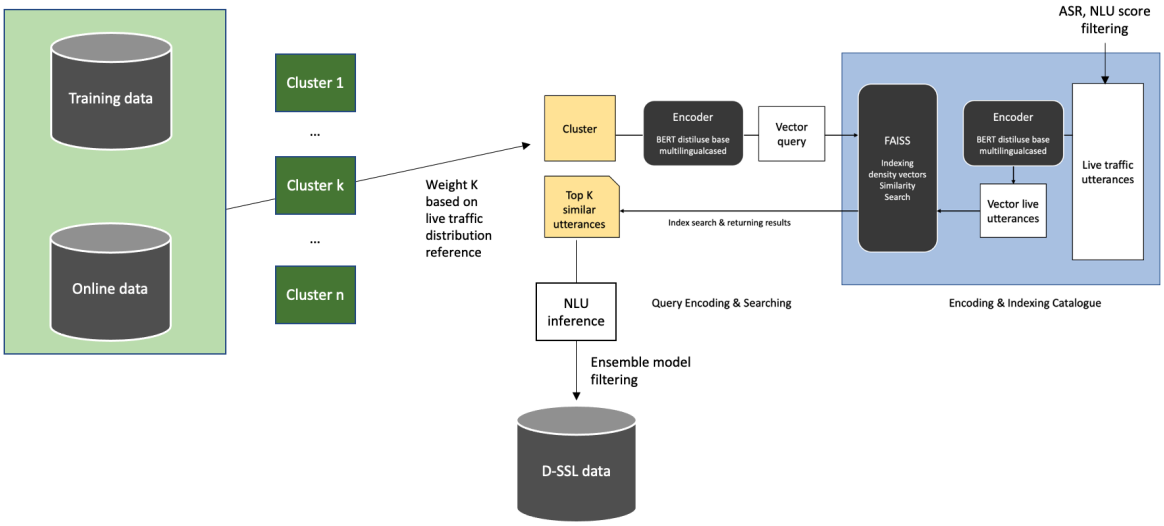

Figure 1: D-SSL framework

Algorithm 2: De-biased SSL data selection

Intput: • Set of unlabelled online traffic data $\mathcal{U} = \{x_i\}_{i=1}^{U}$; $x_i \in \mathbb{R}^n$, n is the embedding size
• Confidence score threshold τ_{NLU}
• Data shift model $S = \{(c_k, w_k)\}_{k=1}^{K}$
Output: $E_{DSSL} = \{(x_i, y_i)\}_{i=1}^{E} | x_i \in \mathcal{U}$
for *each* $k = 1, \ldots K$ **do**
  Calculate average online embeddings for cluster k: $\mathcal{E}_k = average(x_i | x_i \in c_k, x_i \in O)$
  $E_{DSSL}^{(k)}$ = top w_k utterances $\{x_i | x_i \in \mathcal{U}, s_i \geq \tau_{NLU}\}$ that are the closest to $\mathcal{E}_k$
  for *each* $x_i \in E_{DSSL}^{(k)}$ **do**
    Obtain proxy labels y_i for x_i
  end
  $E_{DSSL} = E_{DSSL} \cup E_{DSSL}^{(k)}$
end

labels for those selected utterances using a teacher model. The main setups of the teacher models and experiments are described in section 4.

3.4 Low confidence semi-supervised learning

Confidence score is one of the main parameters when selecting SSL data, as, on one hand, it helps filter out low confidence (and hence presumably incorrect) hypotheses that could harm the model performance, and at the same time preserve confident and accurate predictions. It is also frequently combined with other data selection strategies, such as semantic similarity or ensemble filtering to improve data quality. However, confidence-based selection is challenging as removing too much of low confidence data would inevitably reduce the training data variation, thus making the model less robust. On the other hand, ingesting too many high confidence utterances could potentially only bring marginal improvements as the model is already good at predicting those or increase the amount of error if those samples are incorrect. To understand the impact of different confidence-based selection strategies we carry out a targeted study using different confidence thresholds and a randomly downsampled baseline to eliminate the effect of different data distributions here, in contrast to the D-SSL study (see 3.3). In particular, we aim at selecting low confidence utterances that would increase the data variation, at the same time ensuring that their labelling is correct. To achieve that, we retrieve automatically labeled live traffic data, and for every utterance with a confidence below a certain threshold, we look for other utterances that are similar but have a high confidence in their classification. The similarity of utterances is defined based on the edit distance (see 5).

3.5 Masked Language Model data augmentation

In addition to the previously described semi-supervised approaches, we also experimented with synthetic data generation.

We used the standard Masked Language Model (MLM) objective for pretraining of large language models [Devlin et al. 2019] and modified it to be used for data augmentation.

The general idea of masked language models is to predict tokens that were removed from the input to reconstruct the original sentence before corruption.

In our augmentation scheme we followed the idea of [Ng et al. 2020] to use a MLM at inference time to generate novel completions of the data. They describe the application on different classification tasks as natural language inference and sentiment analysis. We

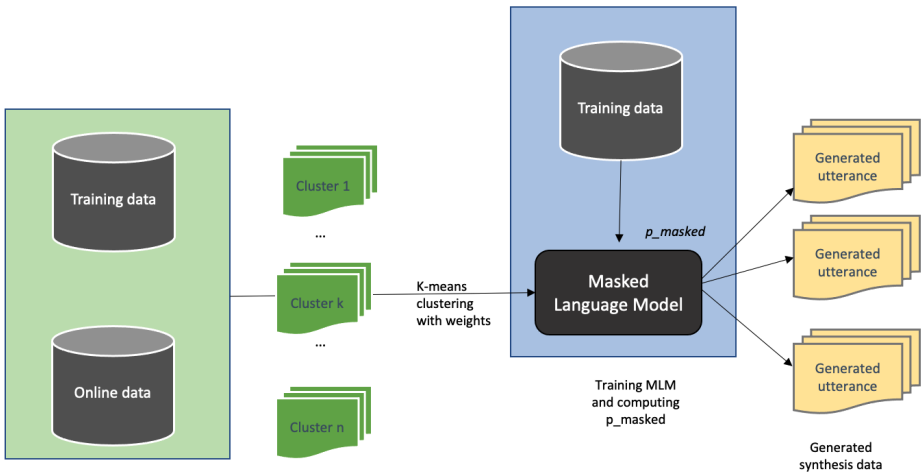

Figure 2: Targeted Masked Language Model data generation

extend their approach to be also applicable to sequence labelling as is the case in natural language understanding. To achieve this, we first finetuned the pretrained models in another training round of MLM training on our existing training data with a combination of the utterance text and the annotation including slots:
Utterance text: set the alarm at nine thirty am <sep>
Domain & intent: Notifications SetAlarmIntent
Named entities: Other Other Other Other Other Time Time

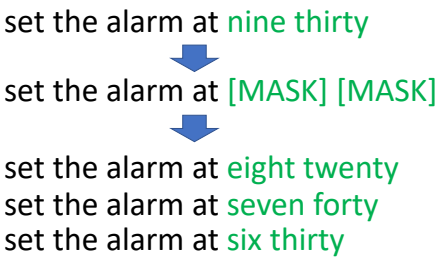

Figure 3: Example of MLM utterance generation with a seed utterance.

For our use case, we want the model to see not only the tokens, but also the annotation, since it might influence how the masked token should be filled in. During this finetuning step tokens are masked with constant probability (0.15; same as during original BERT pretraining objective).

However, during inference, we hypothesise that masking all words with constant probability is not optimal. For example, in the above sentence, "nine" and "thirty" should be masked and replaced with higher probability compared to "alarm", in order to generate many high quality utterances. Thus, we propose a **variable** mask probability method at inference time, in which we mask each word with different probability. The main idea is to find words that should be masked with higher probability (i.e., changing these slots/words are easier and making more sense). For this, we define a metric to measure how *replaceable* each word is within an intent: the number of times that two pairs of sentences in this intent differ only on this word. Then, we assign each word a $maskProb_{word}$ proportionally to this $replaceability$ metric, between a $min_{maskProb}$ and a $max_{maskProb}$. We set these values to $min_{maskProb} = 0.1$, and $max_{maskProb} = 0.5$.

For example, for the toy dataset:

- *turn the light off*
- *turn the tv off*
- *tv off*
- *light off*

light and *tv* are have a $replaceability$ value of 2, which will correspond to these words being masked with probability $max_{maskProb} = 0.5$. All other words are masked with $min_{maskProb} = 0.1$. For example, on a larger German dataset, we find that "thirty" is masked with $max_{maskProb} = 0.5$ in SetAlarmIntent. Further evaluation of this method can be found in section 5.

During inference we then generated new augmented data by selecting input words to be replaced according to the above probability distribution and then selecting the top_n reconstruction. In the case of utterances where multiple words were masked and reconstructed, we made the naive assumption of independent reconstructions of the individual words and selected the top_n reconstructions according to the combination of the individual word reconstructions.

Since this masked language model approach was based on a large model that was pretrained on a similar task but different data, this model was able to create word substitutions that have not been seen in the rest of the training data and therefore introduced more variability into the augmentation data.

By the selection of the seed data to extend and control over the number of reconstructions to be created for every input utterance, this approach allows a guided enrichment of the training data with

examples similar to those that were identified as being underrepresented in the overall data. An example of utterance generation from a seed example can be found in Figure 3.

3.6 Data augmentation filtering with ensemble models

With our data enrichment methods, there is a vast amount of data that we could add to the model. However, there is a risk of introducing wrong annotations which decrease accuracy. To reduce a risk that this happens, we add a filtering step.

The objective is to add correct annotations to the training data. A simple way of achieving this would be to simply select annotations in which the model has high confidence, however there is a risk that, if you only add annotations in which the model is already confident, we cannot improve the generalization capability of the model.

We use the idea of both [Pleiss et al. 2020] and [Laine and Aila 2016] of using the "opinion" of the model at different epochs to produce high quality annotations that also have variability. In the first epochs, the model will be simpler and have worse accuracy compared to the final epochs. We select utterances for which a model trained on fewer epochs and less data (student) agrees with the optimal-accuracy model (teacher; trained the full number of epochs on all data) that an utterance's annotation is likely. Specifically, the condition is

$$\min_{x \in intent, slots} \hat{P}(x) > threshold$$

Where (intent, slots) refer to the intent and slots of the utterance as annotated by the teacher model, and $\hat{P}$ refers to the probability that the student model estimates for the utterance (or output of the softmax function at the output layer of the model corresponding to the intent/ slots). In fact, a high value of this metric will correspond to a high value of the margin metric in [Pleiss et al. 2020], since when a a logit is much higher than all other logits (high margin metrics), after applying softmax function, the probability corresponding to that logit will be high. We choose to use $\hat{P}$ instead of the margin metric for interpretability reasons.

Our model does not produce a sole annotation for each data point (utterance). Rather, it outputs a predicted intent and various predicted slots, each of them with a different $\hat{P}$ value. To aggregate them into a single metric to test against a threshold, several options would be possible, such as the mean. However, we choose the minimum to avoid that wrong slots/ intent with very low $\hat{P}$ do not show in our metric in the case that the sentence has many slots that are predicted with high confidence.

Finally, we come to the threshold. In all our experiments we chose 0.5, which can be interpreted as that the student model thinks the slot/ intent is more likely to be correct than to not be correct. Fixing this value allows us to perform experiments, using both public and commercial datasets, in Portuguese, German and English, without having to do expensive hyper-parameter tuning in every experiment (see section 4).

4 EXPERIMENTAL SETUP

In order to measure the effectiveness of the mitigation techniques, we run the data enrichment framework with both synthetic data augmentation and SSL, comparing with models where no additional data is added. We conduct the experiments on two large voice assistant commercial datasets, in German and Portuguese languages, and measure the performance of each method on a general offline test set as well as through online experiments.

Additionally, for synthetic data generation with MLM, we run experiments on two public NLU datasets SLURP [Bastianelli et al. 2020] (a spoken language understanding resource dataset with 18 domains) and ATIS [Hemphill et al. 1990] (the spoken language systems pilot corpus), where we show how our approach could help to mitigate the data bias in a low data regime setup. For SSL, due to the limited number of data samples in the public datasets, it is not straightforward to simulate the SSL data selection from unlabelled data to study its effectiveness. Our main assumption for the SSL data selection approach is based on the availability of a vast amount of online live traffic data.

4.1 Metrics

In the commercial dataset, we report the following metrics: Semantic Error Rate (*semErr*), Intent Classification Error Rate (*intentER*), and Recognition Error Rate (*recognitionER*). These metrics are highly correlated to each other. *SEMER* is a metric to evaluate jointly Intent Classification and Slot Labelling [Gaspers et al. 2021]. It is calculated as:

$$semER = \frac{\text{\#slot + intent errors}}{\text{\#slots in reference + 1}} \tag{2}$$

On the other hand, *recognitionER* is an utterance-centric metric. An utterance is considered incorrectly labelled if either the intent or any of the slots is incorrect. Then, *recognitionER* corresponds to the fraction of incorrectly labelled utterances:

$$recognitionER = \frac{\text{\#Incorrect interpretation (intent or slot)}}{\text{\#Total utterances}} \tag{3}$$

We also consider a metric to evaluate only the intent accuracy, without considering the slot labelling task. We calculate the fraction of utterances with incorrect intent:

$$intentER = \frac{\text{\#Incorrect intent}}{\text{\#Total utterances}} \tag{4}$$

4.2 Experiments on commercial datasets

As often observed in real world applications, the problem of bias in training data and content shifting can become quite severe, especially when the functionalities and domains are frequently evolving. We measure the impact of our framework for two different languages, German and Portuguese.

Datasets We evaluate our method on two commercial voice assistant-related datasets: one in German, and one in Portuguese language (with the Brazil dialect). All utterances are de-identified prior to experimentation. The size of the test data for Portuguese is more than 500K utterances, and for German test set, it contains over 1M utterances.

Model We run a joint intent classifier and named entity recognizer on a large amount of commercial training data. We use a common spoken language understanding (SLU) architecture based on pre-trained BERT models. It consists of a BERT encoder, an intent and a slot decoder. In particular, given an utterance, the BERT

encoder outputs both at token and sentence level. These encoders are used as inputs for the intent and slot decoders. For the intent decoder, we use a standard feed-forward network with two standard dense layers and a softmax layer on top to classify the utterances into a pre-defined set of intents. For the named entity recognized (slot decoder), it uses a conditional random field layer on top of the two dense layers. During training, the loss of both the intent classifier and the named entity recognized are jointly optimized with an equal weight.

4.3 Experiments on public datasets

Finally, in order to evaluate the performance of the data enrichment method, we run MLM synthetic data generation on two public datasets. For SSL, we assume that there will be a large amount of online/unlabelled data to select from. Due to this assumption, it is more tricky to replicate the experiments on these public datasets due to the limited amount of data.

Data and experimental setup: We use two public NLU datasets which are annotated both with slots and intent: SLURP and ATIS. In both datasets, we combine train, test and dev data, and create new splits: 50, 40 and 10 % for test, train and dev data respectively. The high proportion of test data simulates a low data setting, but also reduces the variance of the performance metrics.

Models: In this experiment we use the same pre-trained BERT model [Devlin et al. 2018] for generating MLM data, for filtering the data as in 3.6, and for evaluating performance: the *bert-base-uncased* model from [Wolf et al. 2019]. We train this model over 4 epochs, with fixed learning rate 3e-5 and batch size 32. To calculate each word's mask probability according to the variable probability approach described in section 3.5, we consider each 4-gram to be its own utterance (due to the smaller amount of data here compared to the commercial dataset setting). We also manually tune the mask probability for one-word slots, which we mask with probability $max_{maskProb} = 0.5$. We iterate through the data twice, generating every time 3 utterances/ masked utterance. Thus, pre-filtering we increase training data size x6.

5 RESULTS

5.1 Experiments on the commercial datasets

The results of the models with relative differences are reported in Table 1 for both techniques – data augmentation via D-SSL and synthetic data generation via MLM. We report domains where we observed most degradations and improvements. All results are reported with respect to the baseline model, where no mitigation technique is applied.

Overall, we observe that our data augmentation techniques help overcome bias in the training data in most domains. As one can see from the table, models trained with each individual technique outperform the baseline by 4.59% relative SEMER for German and 1.59% relative SEMER for Portuguese. We also find that each technique is better for a different set of domains: For example, in German synthetic data outperforms D-SSL in Daily Briefing and Home Automation, while D-SSL outperforms synthetic data overall for Portuguese.

Our hypothesis is that synthetic data generation can generate novel annotated utterances with a reduced risk of confirmation bias (model sees its own errors during training) compared to D-SSL. The side effect is that synthetic data generation might create unnatural utterances while D-SSL selects real utterances that have been asked by customers. Depending on the domain, one method will get better results than the other and by combining techniques, we expect better performance since they will complement each other to address domains peculiarities. Additionally, both languages show that combining D-SSL and synthetic data generation with data distribution re-weighting is important to obtain good results, as the re-weighting detects which patterns of utterances are missing from the training data and should be added.

	German		Portuguese	
Domain	MLM	D-SSL	MLM	D-SSL
DailyBriefing	-29.92	-6.42	0.04	-0.61
Global	0.17	-7.27	-2.62	-2.1
HomeAutomation	-9.39	-5.66	-6.77	-4.24
Music	1.22	0.79	-1.7	-4.2
Notifications	2.31	-3.63	0.16	0.33
Todos	-10.5	-10.3	4.46	3.9
Recipes	-21.23	-12.67	-0.19	0.32
Sport	-22.09	-3.76	-6.59	-5.28
GeneralMedia	-8.46	-3.56	2.38	-4.02
Overall	**-4.59**	**-4.59**	**-1.59**	**-1.74**

Table 1: Semantic error rate (rel. change) for synthetic data and D-SSL experiments

Since the D-SSL approach only selects utterances with high confidence scores, we further experiment how selecting low-confident utterances would help to mitigate the data unbalance problem. In this experiment, we define high confidence as being similar, if the confidence assigned to the utterances labels by the classifier is above a certain threshold X and if the edit distance (including domain, intent and slots) is less than N. We experimentally define the threshold $X = 0.7$ and the edit distance $e = 1$. Note that we also remove utterance with confidence 1, as those are covered by a rule-based component of our system and thus do not add any value. We also remove any repeating utterances. To reduce the search space, we split the data by domain and sentence length, and use reverse indexing to store the information about word occurrences within certain sentences, which speeds up the candidate lookup. We used the following confidence-based selection strategies:

(1) **High-confidence SSL:** Utterances above the defined threshold X.
(2) **Low-confidence SSL:** Utterances below the defined threshold and with edit distance less than N.
(3) **Combined SSL:** Using utterance from both (a) and (b).

The results of the low-SSL experiments are reported in Figure 4. We report the results on the same domains as in the previous experiments. The result suggests that utterances with highest confidence are less beneficial for the model performance for most individual domains as those are already well recognized by the model and do not add further improvement, but rather amplify the model error. Compared to that, the ingestion of low-confidence and combined approaches helps bring more variation into the data and improve

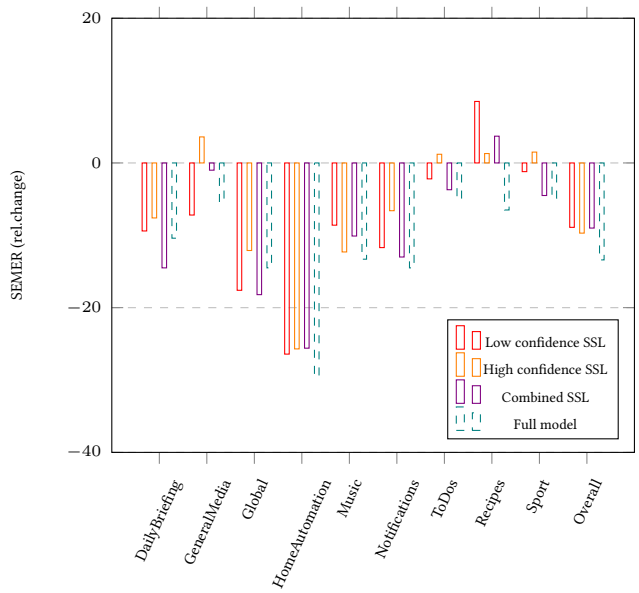

Figure 4: Comparison of confidence-based SSL data selection. The graph presents rel.change in SEMER with respect to a low data baseline. Note that lower values represent better performance.

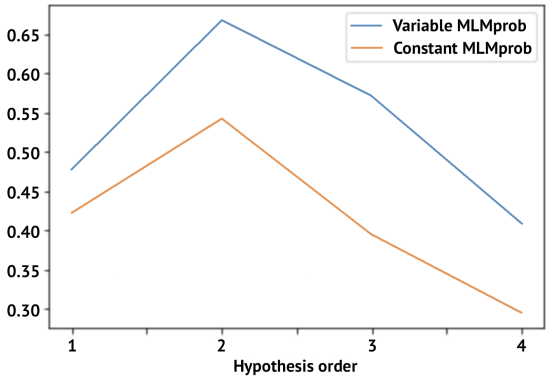

Figure 5: Fraction of generated utterances that occur in the dev set. We note the second hypothesis has the highest fraction in the dev set, which indicates high quality of the generated data.

	German	Portuguese
SSL without data shift model	+1.26	+0.46
SSL without post-filtering	+3.28	+1.52

Table 2: Semantic error rate (rel. change) for SSL experiments without data shift model and without ensemble model filtering

performance for 7 out of 10 domains. For Music, interestingly, selecting utterances with highest confidence yields best result. The combination of both approaches even helps outperform the full model for DailyBriefing and Global domains. For HomeAutomation, all three selection strategies show comparable result (with low-confidence SSL being slightly better than the other two approaches). We also observe degradation for one domain, Recipes, which is one of the smallest domain and is this probably most affected by the low data regime. For the overall model, we see that all approaches show a rather comparable performance.

Impact of variable mask probability in MLM: Before adding the MLM generated data to the training data to observe its impact on downstream metrics such as semER, we want to do a quality check of the generated data, as well as observe the isolated impact of the **variable** mask prob method described in section 3.5. In this experiment, we split the data in training (75%) and dev (25%). We want to investigate what fraction of the generated utterances is in the dev set: since these utterances are not part of the MLM training data, being able to generate them would indicate high quality of the data. In Figure 5 we note that using our proposed variable mask probability approach increases the fraction of generated utterances in the dev set. We group utterances by their hypothesis order (based on the probability in the output layer, n-best hypothesis are generated, ordered by score). We use this investigation to choose n=3, since at higher hypothesis orders the quality of the generated utterances decreases. The optimal generated utterances occur at order two, instead of (as might be expected) one: this is because the first hypothesis is more likely than subsequent ones to replicate exactly a training utterance (which do not appear in the dev set).

Impact of the data shift model: In order to learn the impact of the data shift model, we ran an experiment where the SSL data is randomly selected based on the online live traffic distribution, without taking into consideration the difference between training vs. online live data as described in algorithm 1. The SemER relative changes are reported in table 2, where we see that selecting SSL without using the data shift model calculation would even result in degradations for both German and Portuguese (+1.26% and +0.46% degradation in compared to models without the added data). It has clearly shown that the weighting metrics using the data shift model could help to select the right data to improve the model performance.

Impact of the ensemble filtering model: During the experiments, it has been shown that having a filtering model on top of the data selection is crucial in the data enrichment framework. In fact, when running data augmentation and SSL without running the data filtering at the end, we observe a decrease in the model performance (table 2) with +3.28% for German and +1.52% for Portuguese. These numbers in compared to the overall relative changes reported in table 1 have clearly shown that the data shift model together with post-filtering are important in order to get the overall improvement for both models German and Portuguese (-4.59% and -1.74% semantic error rate for two models respectively).

5.2 Experiments on public datasets

As can be seen in table 3, adding the MLM data without any filtering actually slightly degrades the performance of the models. However, after adding the filtering step, the best results are obtained, with improvements of up to 44% in intentER in the ATIS dataset. These results show that reusing BERT's original pretraining objective to generate variations of the training data can produce high quality utterances that reduce error rates.

	recogER	intentER	SLOT Error rate
ATIS, baseline	0.117	0.033	0.013
ATIS, MLM	0.127	**0.07**	0.014
ATIS, MLM + filtering	**0.090**	**0.018**	**0.01**
SLURP, baseline	0.360	0.171	**0.065**
SLURP, MLM	0.40	0.177	0.074
SLURP, MLM + filtering	**0.34**	**0.157**	0.06
ATIS IMPROVEMENT	-22.97	-44.46	-20.69
SLURP IMPROVEMENT	-5.61	-8.07	0.17

Table 3: Results of adding MLM data to public datasets on a low data setting

6 CONCLUSIONS

In this paper, we have described a framework to mitigate the problems of unbalanced data distribution in online vs. training data in real world applications. The data enrichment framework is based on the calculation of the data drift model, where we estimate the difference between online and training data using k-means clustering with a weighting function. We explain how the debiased semi-supervised learning model together with synthetic data augmentation can be used together with post-filtering functions. We have conducted various experiments to show how these techniques are practically applied to improve the performance of spoken language understanding systems. We have further done an analysis of semi-supervised learning performed on different types of data from high to low confidence data selection. It is shown that the data shift model and the ensemble filtering model are crucial in order to achieve a good performance in the data enrichment framework.

REFERENCES

Ehsan Amid, Manfred K. Warmuth, Rohan Anil, and Tomer Koren. 2019. Robust Bi-Tempered Logistic Loss Based on Bregman Divergences. *CoRR* abs/1906.03361 (2019). arXiv:1906.03361 http://arxiv.org/abs/1906.03361

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. SLURP: A Spoken Language Understanding Resource Package. *CoRR* abs/2011.13205 (2020). arXiv:2011.13205 https://arxiv.org/abs/2011.13205

David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. *CoRR* abs/1905.02249 (2019). arXiv:1905.02249 http://arxiv.org/abs/1905.02249

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning. *IEEE Transactions on Neural Networks* 20, 3 (2009), 542–542.

Eunah Cho, He Xie, and William M Campbell. 2019. Paraphrase generation for semi-supervised learning in NLU. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. 45–54.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*. 4171–4186.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A Survey of Data Augmentation Approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 968–988. https://doi.org/10.18653/v1/2021.findings-acl.84

Judith Gaspers, Quynh Ngoc Thi Do, Daniil Sorokin, and Patrick Lehne. 2021. The impact of intent distribution mismatch on semi-supervised spoken language understanding. *Interspeech* (2021).

Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou (Eds.), Vol. 17. MIT Press. https://proceedings.neurips.cc/paper/2004/file/96f2b50b5d3613adf9c27049b2a888c7-Paper.pdf

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. 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*. https://aclanthology.org/H90-1021

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *CoRR* abs/1702.08734 (2017). arXiv:1702.08734 http://arxiv.org/abs/1702.08734

Sosuke Kobayashi. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 452–457.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. 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)*. 3609–3619.

Samuli Laine and Timo Aila. 2016. Temporal Ensembling for Semi-Supervised Learning. *CoRR* abs/1610.02242 (2016). arXiv:1610.02242 http://arxiv.org/abs/1610.02242

Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* (2020).

Qimai Li, Xiao-Ming Wu, and Zhichao Guan. 2019. Generalized Label Propagation Methods for Semi-Supervised Learning. *CoRR* abs/1901.09993 (2019). arXiv:1901.09993 http://arxiv.org/abs/1901.09993

Lu Liu and Robby T Tan. 2021. Certainty driven consistency loss on multi-teacher networks for semi-supervised learning. *Pattern Recognition* 120 (2021), 108140.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah M. Er ni, and James Bailey. 2020. Normalized Loss Functions for Deep Learning with Noisy Labels. *CoRR* abs/2006.13554 (2020). arXiv:2006.13554 https://arxiv.org/abs/2006.13554

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1268–1283.

Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2021. Towards robust learning with different label noise distributions. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 7020–7027.

Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. 2020. Identifying Mislabeled Data using the Area Under the Margin Ranking. *CoRR* abs/2001.10528 (2020). arXiv:2001.10528 https://arxiv.org/abs/2001.10528

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *CoRR* abs/1908.10084 (2019). arXiv:1908.10084 http://arxiv.org/abs/1908.10084

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)*. 86–96.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685* (2020).

Antti Tarvainen and Harri Valpola. 2017a. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780* (2017).

Antti Tarvainen and Harri Valpola. 2017b. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR* abs/1703.01780 (2017). arXiv:1703.01780 http://arxiv.org/abs/1703.01780

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR* abs/1910.03771 (2019). arXiv:1910.03771 http://arxiv.org/abs/1910.03771

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised Data Augmentation. *CoRR* abs/1904.12848 (2019). arXiv:1904.12848 http://arxiv.org/abs/1904.12848

Xueyuan Zhou and Mikhail Belkin. 2014. Semi-supervised learning. In *Academic Press Library in Signal Processing*. Vol. 1. Elsevier, 1239–1269.

Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning* 3, 1 (2009), 1–130.

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